
Universidade de Aveiro
2009
Departamento de Electrónica, Telecomunicações e Informática

Eduardo Manuel Moreira Seabra

Gestão de Identidades e privacidade em redes de próxima geração

Universidade de Aveiro
2009
Departamento de Electrónica, Telecomunicações e Informática

Eduardo Manuel Moreira Seabra

Gestão de Identidades e privacidade em redes de próxima geração

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. Francisco Manuel Marques Fontes, Professor Auxiliar Convidado do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

Dedico este trabalho à minha família e namorada pelo incansável apoio.

o júri

Presidente

Prof. Doutor José Carlos da Silva Neves
Professor Catedrático da Universidade de Aveiro

Prof. Doutor Paulo Alexandre Ferreira Simões
Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Prof. Doutor Francisco Manuel Marques Fontes
Professor Auxiliar Convidado da Universidade de Aveiro

agradecimentos

Agradeço ao meu orientador por me ter dado a oportunidade de realizar esta dissertação que faz uso de conceitos e tecnologias de extrema actualidade, interesse e aplicabilidade. Agradeço ao meu co-orientador Eng. Ricardo Azevedo Pereira por toda ajuda e apoio. Agradeço ao Instituto de Telecomunicações e ao Departamento de Electrónica e Telecomunicações da Universidade de Aveiro pelos recursos disponibilizados, e agradeço finalmente e não menos importante, à minha família, namorada, colegas e amigos por todo o apoio.

palavras-chave

Identidade, identidade virtual, identificadores, gestão de identidade, privacidade, confiança, segurança, provedor de identidade, provedor de serviço, federação.

resumo

Esta Dissertação de Mestrado Integrado faz um estudo das normas e tecnologias existentes sobre Gestão de Identidades e idealiza novos cenários de aplicação. Descreve um pequeno protótipo de um provedor de identidades que inclui algumas novidades para a implementação dos cenários propostos desenvolvido no de decurso desta dissertação. Apresenta também algumas soluções de como implementar Gestão de Identidades em Redes de Próxima Geração.

keywords

Identity, virtual identity, identifiers, identity management, privacy, trust, security, identity provider, service provider, federation.

abstract

This work is a study of existing standards and technologies on Identity Management and idealize new scenarios of application. Describes a small prototype of a identity provider, that includes some new features for the implementation of the proposed scenarios developed in the course of this dissertation. Also presents some solutions for how to implement Identity Management in Next Generation Networks.

Índice

Lista de Figuras

1 Introdução

1.1 Enquadramento do trabalho

A Internet continua a ser cada vez mais valiosa, enfrentando continuamente novos desafios. Estes desafios passam por proteger a identidade, a privacidade de cada um dos seus utilizadores. À medida que a Internet cresce, novos serviços surgem, mais contas, *passwords*, perfis e outro tipo de informações que têm de ser geridas e guardadas por cada um desses utilizadores. Acontece que todos temos limites e não conseguimos gerir toda esta vastíssima informação, o que nos leva a cometer erros, o que, por exemplo, faz com que usemos as mesmas *passwords* em variadíssimos sites, podendo muitos não ser os confiáveis, o que poderá trazer consequências irremediáveis para as nossas vidas. O ideal seria que alguém, uma entidade ou um serviço que fosse da nossa inteira confiança, fizesse toda esta gestão por nós – fizesse a **Gestão de Identidades**.

1.2 Objectivos

Esta Dissertação, executada no âmbito do Mestrado Integrado do Curso de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro, tem como principais objectivos:

- conhecimento das normas e tecnologias relacionadas com Gestão de Identidades,
- a criação de cenários de aplicação que tivessem o Operador de Telecomunicações como elemento central para a Gestão de Identidades, e daí identificar novas oportunidade de negócio,
- a criação de um pequeno protótipo que pudesse demonstrar um dos cenário anteriores e
- encontrar extensões aos mecanismos A4C (Autenticação, Autorização, Contabilização, Tarifação e Auditoria) para proporcionar o suporte de Gestão de Identidades em redes de próxima geração.

1.3 Estrutura da Dissertação

A Figura 1 apresenta a estrutura da Dissertação. A Dissertação começa com uma descrição de conceitos (Capítulo 2) necessários para uma melhor compreensão de todo o tema; o que é a nossa identidade, elementos que a compõem, privacidade, etc. Continua com o Estado da Arte (Capítulo 3), para no Capítulo seguinte se idealizarem alguns cenários de aplicação (Capítulo 4) usando o que foi aprendido. O Capítulo seguinte descreve a implementação de um dos cenários idealizados no Capítulo 4, tendo para isso sido desenvolvido um pequeno, mas eficiente, servidor para a Gestão de Identidades. No penúltimo Capítulo faz-se um estudo dos elementos das redes de próxima geração responsáveis pelo A4C, de forma a integrar a Gestão de Identidades. No último Capítulo faz-se uma pequena reflexão sobre o que foi estudado e implementado, e também sobre os pontos ainda menos bem definidos e a melhorar para futuro.

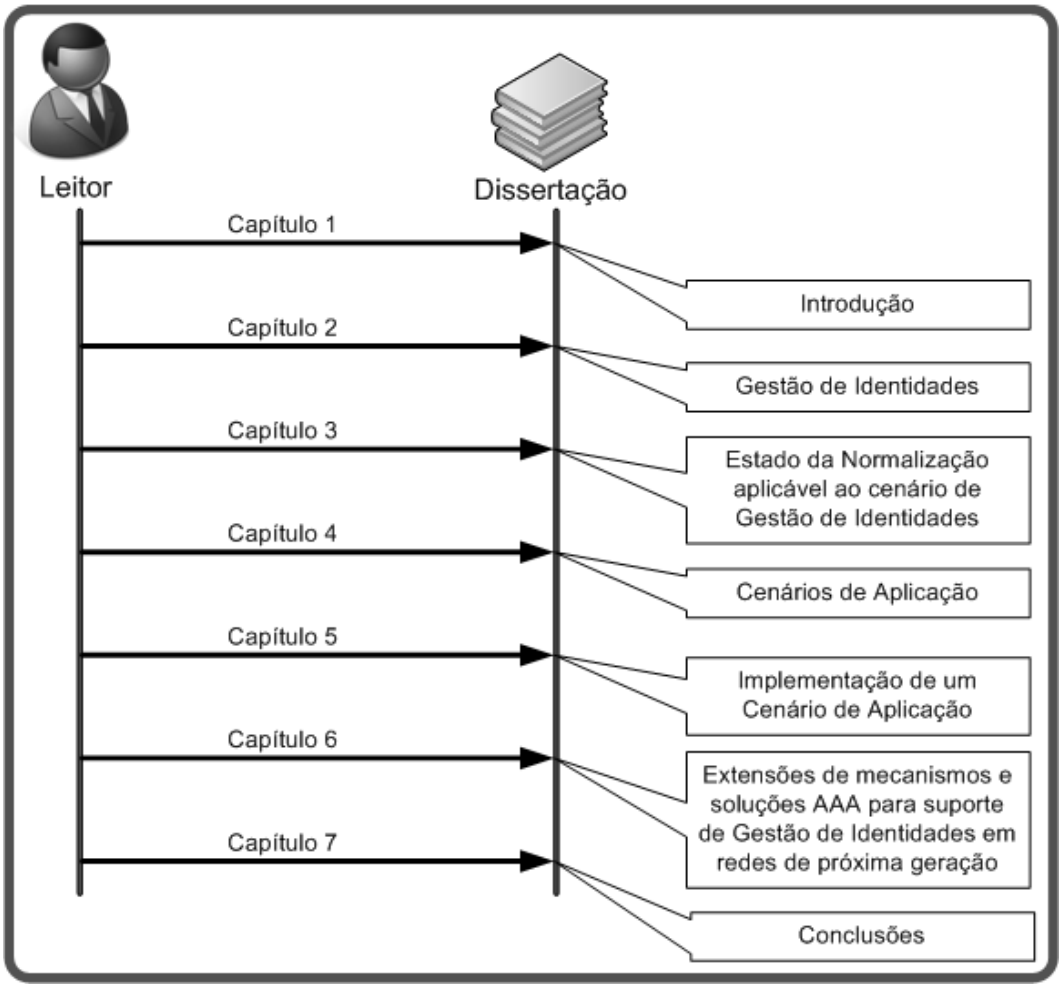


Figura 1 – Estrutura da Dissertação de Mestrado Integrado sobre Gestão de identidades e privacidade em redes de próxima geração

2 Gestão de Identidades

2.1 O que é a Identidade?

Identidade, segundo o dicionário da Língua Portuguesa da Porto Editora, é o "conjunto de características (físicas e psicológicas) essenciais e distintivas de alguém, de um grupo social ou de alguma coisa".

A identidade de alguém é **quem** ela é ou o **que** ela é. Esse alguém, como o significado acima diz, é o conjunto de características essenciais e distintivas, e a estas características vamos chamar **atributos**. Um atributo, pode ser o seu nome, a sua data de nascimento, a cor dos seus olhos, a sua altura, o seu peso, a sua profissão, entre outros.

Um indivíduo pode ser visto como alguém que é, que faz, que sabe ou que tem algo.

Um indivíduo também é o conjunto de todos os atributos que familiares, amigos, conhecidos e autoridades têm sobre ele.

Quando se conhece alguém pela primeira vez, normalmente é trocada alguma informação sobre a identidade de cada um. Mas também se pode saber mais sobre esse indivíduo ouvindo o que os outros dizem acerca dele - **reputação**. Na maior parte das vezes o que alguém é, é uma combinação dos seus atributos e da sua reputação.

*James Kobielus, um analista Industrial, diz no seu blog**Error! Reference source not found.** que a reputação não é uma identidade, credencial, permissão ou papel. Mas também não é exactamente um atributo, no mesmo sentido que se diz que é um atributo a data de nascimento ou a cor dos olhos. Também não é algo sobre o que alguém possa reclamar alguma protecção e privacidade – é exactamente o oposto: o tribunal da opinião pública sobre o qual não existe o mais pequeno controlo e soberania. No contexto Gestão de Identidades, a reputação é mais do que uma garantia ou nível de confiança – é uma avaliação da medida em que vale a pena conhecer ou associar-se a alguém.*

A Identidade de um indivíduo é, portanto, a combinação de dois aspectos, uma identidade subjectiva e uma objectiva/social (reputação) [75], isto é, Identidade é algo que um indivíduo é (características físicas e factos sobre ele) e algo que os outros têm sobre ele.

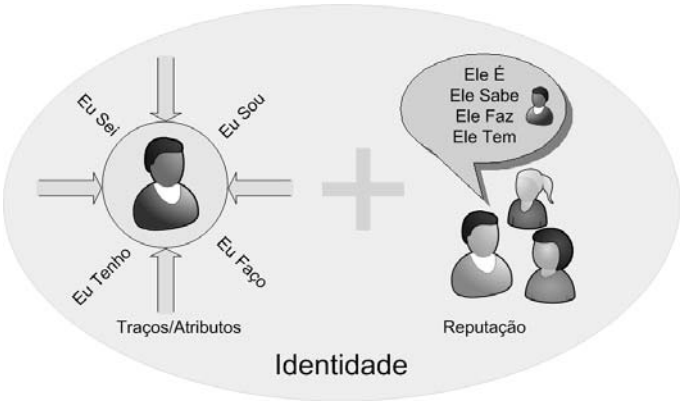


Figura 2 – Identidade é a soma da identidade subjectiva com a identidade objectiva/social (reputação)

2.2 Prova da Identidade e Corrente de Confiança

Muitas vezes é pedido a um indivíduo que prove a sua identificação. Existem várias formas de um indivíduo o fazer, por exemplo, através da sua assinatura, da sua impressão digital, ou apresentando um documento de identificação.

Por exemplo: um indivíduo vai a conduzir o seu automóvel e é mandado parar por um polícia. Este pede-lhe a carta de condução.

- A corrente de confiança começa quando a Autoridade competente para a emissão da carta de condução, verifica que o indivíduo (condutor) tem competência e capacidades para conduzir determinado tipo de veículo e que este vai respeitar o Código da Estrada.
- O condutor do veículo mostra ao polícia a sua carta de condução.
- O polícia vê a carta de condução e confia nela, que diz que o indivíduo está habilitado a conduzir aquele veículo. E confia nela porque essa carta de condução foi emitida pela Autoridade competente para o efeito, após o indivíduo ter cumprido os requisitos exigidos por esta Autoridade.

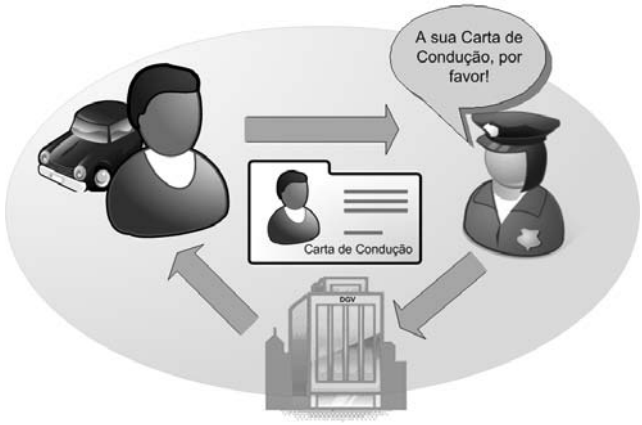


Figura 3 – Quando o condutor mostra a carta de condução ao polícia, é criada uma corrente de confiança

2.3 Camadas da Identidade

Até agora falou-se aqui de Identidade no singular mas, de facto, um indivíduo tem múltiplas identidades. Estas múltiplas identidades não são mais que diferentes faces da Identidade. Para uma determinada entidade, a identidade de um indivíduo pode ser apenas um determinado conjunto dos atributos deste. Por exemplo, o conjunto de atributos que o banco vê de um indivíduo são os números das suas contas bancárias ou os números dos seus cartões de crédito, entre outros. Se for a entidade empregadora desse indivíduo, deverá ter acesso a outro conjunto de atributos, como por exemplo o número da segurança social e o número da conta bancária para efectuar a transferência do vencimento.

Figura 4 - As múltiplas identidades de um indivíduo

Estas múltiplas identidades estão todas ligadas pelo indivíduo. São apenas diferentes perspectivas do mesmo indivíduo e dos atributos que ele possui. Estas múltiplas identidades, ou ***personas***, estão todas ligadas por pequenos elementos comuns, as chaves, que usualmente são: o nome, o endereço, a data de nascimento e o número do bilhete de identidade.

Andre Durand, fundador e Director-Executivo da *Ping Identity Corporation* [2], introduziu o conceito de "*Tiers of Identity*" (Camadas da Identidade) – Figura 5.

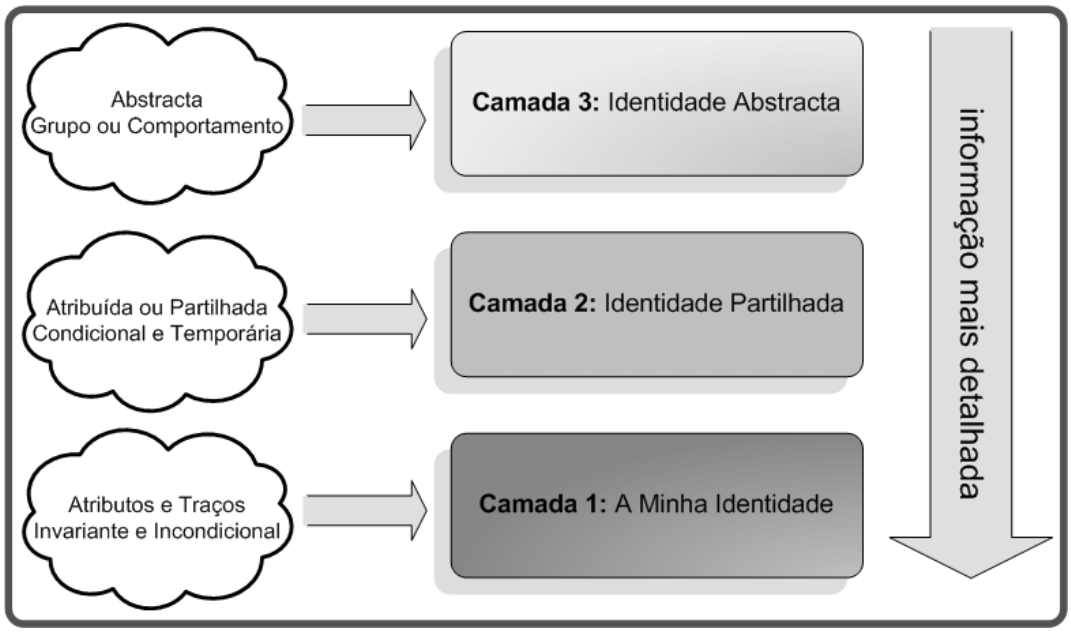


Figura 5 – As três camadas da Identidade

A ***camada 1*** é chamada de "A Minha Identidade". Consiste nos atributos e traços associados a um indivíduo. São elementos que não se vão modificar com o tempo e são incondicionais, como por exemplo a cor dos olhos, o nome, a data de nascimento, entre outros.

A ***camada 2***, chamada de "Identidade Partilhada", consiste nos atributos atribuídos por outros. Estes atributos são partilhados porque são atributos usados para identificar mas são temporariamente atribuídos baseados em algum tipo de relação. A carta de condução, o número do cartão de crédito e o cartão do clube de golf, são exemplos de informação que pode ser atribuída a um indivíduo. Quando a relacionamento que define um daqueles elementos é terminada, o atributo associado deixa de ser válido.

A ***camada 3***, chamada de "Identidade Abstracta" estabelece a identidade de grupos. Por exemplo, um indivíduo pode ser identificado como estudante, como sendo maior de idade ou como membro de qualquer outro tipo de grupo demográfico. Outro exemplo é o de uma empresa classificar um indivíduo como um "*cliente Premium*" ou outra empresa classificar o mesmo indivíduo como um "*comprador frequente*". Este tipo de classificações identifica o indivíduo, mas de uma forma abstracta.

As relações da camada 2 usualmente acontecem com o consentimento da pessoa e, na maior parte das vezes, são bem-vindas porque são de benefício próprio. As relações da camada 3 são usualmente forçadas e as pessoas podem ressentir-se delas. Por exemplo, a correspondência comercial não solicitada ou telefonemas comerciais também não solicitados são um problema da camada 3. O problema da camada 3 é que é uma camada pouco fiel e não específica. Consequentemente, a camada 3 raramente pode adequar-se a necessidades reais.

2.4 Identidade Digital

Quase toda a gente, pelo menos numa sociedade civilizada, tem os seus atributos (nome, data de nascimento, sexo,…) registados em algum lado. A sua morada, o seu número de telefone, do telemóvel, a sua ocupação também está registada em algum lado. O Estado, escolas, empresas, associações, também podem ter a identidade das pessoas guardada digitalmente em, por exemplo, bases de dados nos seus servidores.

A **identidade digital** não é mais do que uma parte da identidade de um indivíduo – dados em bits e bytes. A identidade digital tem que ser criada, armazenada, trocada via redes electrónicas, usada, copiada, apagada, manipulada e até revogada. A Gestão de Identidades não é mais do que saber como os utilizadores são identificados, que direitos devem ter, como controlar o seu comportamento e como organizar a sua administração.

A tendência actual é para que entidades das mais diversas áreas (negócios, entretenimento, sociais, etc.) migrem as suas actividades para a Internet. A Internet é um meio primordial para a interacção entre pessoas e organizações. Pela sua natureza, a Internet tem as suas próprias regras. O desenvolvimento de uma regulação para a identidade digital é muito recente. Aplicações e sistemas novos tinham a tendência de criar novas soluções proprietárias de Gestão de Identidades. Nenhuma destas soluções atingiu um nível de uso global que se possa considerar uma norma. Actualmente existe uma colaboração entre vastíssimas organizações e empresas para a adopção de uma norma para a gestão e transferência de identidades. O desenvolvimento de uma norma para a gestão de identidades é essencial para a integração e personalização de *Web Services*.

2.5 Ciclo de Vida

As identidades digitais têm um ciclo de vida, como mostrado na Figura 6. O ciclo de vida de uma identidade digital tanto é aplicado em sistemas de gestão de identidades extremamente complexos como em pequenas contas num computador pessoal em casa.

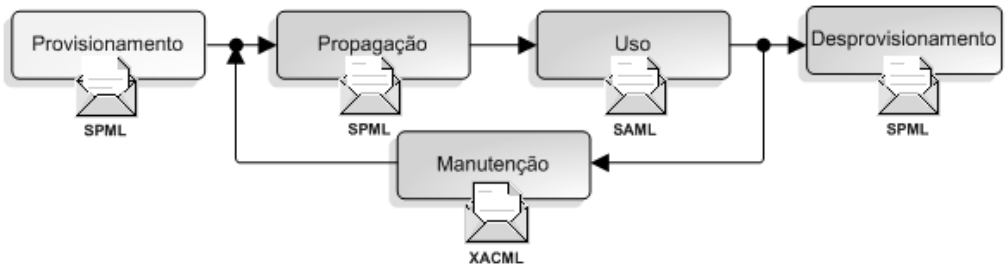


Figura 6 – Ciclo de Vida de uma identidade digital e protocolos associados a cada uma das fases.

Uma identidade digital começa por ser aprovisionada ou criada. Uma vez criada, a identidade digital é propagada para qualquer sistema que a use. Depois de propagada, a identidade pode ser usada. Ocasionalmente, algum tipo de manutenção ou actualização pode ser feita, como alterar a *password*, adicionar um atributo e de seguida volta a ser propagada. Em alguma altura, a identidade digital deixou de servir o propósito para que foi criada e é revogada ou destruída. Uma breve explicação dos protocolos aplicáveis neste processo encontra-se no Anexo 1.

2.6 Características e Conceitos de Gestão de Identidades

2.6.1 Integridade, Não-repúdio e Confidencialidade

Integridade, Não-repúdio e Confidencialidade são princípios básicos na segurança de dados. Um utilizador só usará um sistema de Gestão de Identidades se sentir segurança.

Integridade é dar a garantia de que os dados enviados por uma fonte são os mesmos que são recebidos pelo destinatário e que ninguém está a adulterar essa informação durante o processo.

Não-repúdio é assegurar uma evidência de que a mensagem foi enviada/recebida ou que uma transacção foi iniciada/finalizada. Este processo envia um recibo para cada uma das partes envolvidas no processo [76].

Confidencialidade é um princípio básico em segurança, que assegura que os dados são apenas acessíveis a utilizadores autorizados.

2.6.2 Confiança

Confiança é um conceito implicitamente entendido por um humano. É difícil de quantificar, medir e encontrar um algoritmo apropriado para o implementar. Confiança pode aparecer a diferentes níveis: pessoa a pessoa, pessoa a grupo (pessoas ou entidades), pessoa a sistema, grupo a grupo ou sistema a sistema.

Confiança é um sentimento muito subjectivo. A confiança tem algumas propriedades interessantes:

- Confiança é transitiva apenas em circunstâncias muito específicas
- Confiança não pode ser partilhada
- Confiança não é um sentimento simétrico
- Ser-se confiável (fidedignidade) não é algo que se possa afirmar, ser-se confiável é baseado na reputação.

Em termos de Identidade Digital, confiança não é mais que um conjunto de credenciais e atributos. A fidedignidade é baseada na reputação. A reputação é algo que é gradualmente criado em comunidades de confiança[3]. Este tipo de comunidades é constituído por 5 componentes:

- governação (criam regras, responsabilidades, etc),
- pessoas envolvidas em relações de confiança,
- processos para operações e transacções,
- ferramentas tecnológicas (software e hardware)
- e um modelo económico viável.

Uma infra-estrutura de gestão de Identidades Digitais tem que ser criada e obedecer aos requisitos de todas as características de uma comunidade de confiança[3].

2.6.3 Segurança

Um indivíduo pode ter inúmeras identidades na Internet. Em muitas situações é fundamental garantir que aquele indivíduo, que apresenta uma determinada credencial para a sua identificação, é quem diz ser. O uso da identidade de alguém por um estranho pode ter efeitos devastadores. A protecção da identidade do acesso não autorizado é fundamental para a confiança das pessoas em sistemas de gestão de identidades. Sem segurança o desenvolvimento de Web Services, dentro de uma arquitectura orientada a serviços (SOA – Service Oriented Architecture), é bastante limitado.

2.6.4 Privacidade

Além da sua identidade protegida, as pessoas também querem preservar a sua privacidade. É importante definir qual é o nível aceitável de privacidade para um indivíduo, e o debate é grande, visto que o nível aceitável de privacidade é diferente em diferentes países.

Muitas empresas de *e-business*, para satisfazerem os seus clientes, usam cookies para reconhecer o cliente aquando da próxima visita, registar os seus hábitos, os seus gostos, as suas compras. Usam estas informações para criar um perfil do cliente, perfil este que vai servir para criar ofertas personalizadas, de forma a agradar o mais possível o cliente. O problema é quando esta informação é usada com o intuito de ser vendida a terceiros.

Os clientes têm o direito de saber que dados são recolhidos, porque é que são recolhidos e o que é que as empresas fazem com eles. Também é importante que as empresas usem apenas a informação estritamente necessária para interagir com o cliente. Os negócios na Internet permitem o uso de pseudónimos. Um sistema pode associar um identificador com atributos, direitos e privilégios a uma pessoa da mesma forma como se esta usasse os seus dados correctos (o seu nome por exemplo).

2.6.5 Políticas de Acesso

Um conceito muito importante em Gestão de Identidades é o conceito de Políticas de Acesso. As Políticas de Acesso definem quem pode ou não aceder a determinado recurso. Exemplos de Políticas de Acesso:

- Toda a gente tem permissão para ler a pasta XYZ e os seus ficheiros.
- Todos os utilizadores autenticados podem ler e escrever na pasta XYZ e nos seus ficheiros.
- O utilizador “ABCD” tem permissão para ler e escrever na tabela “Encomendas” da base de dados “Vendas”.

Um dos principais aspectos das políticas de acesso está relacionado com a **Responsabilidade**. Normalmente esta responsabilidade é classificada em três categorias: donos, guardiães e utilizadores.

O dono de um recurso tanto pode ser o seu criador, como outra pessoa que herdou ou a quem lhe foi atribuído esse recurso. O guardião é o gestor do recurso, que tem a responsabilidade de manter a integridade deste, já que é da sua responsabilidade a aplicação correcta das políticas de acesso. O criador e o guardião poderão ser a mesma pessoa ou entidade. Um administrador de um sistema é basicamente um guardião. Um utilizador é alguém ou entidade que quer aceder a um recurso.

Um princípio fundamental nas Políticas de Acesso é o princípio do **Privilégio Mínimo**. Este princípio diz que aos utilizadores não deve ser dado mais acesso aos recursos do que o estritamente necessário para o cumprimento das suas funções. Este princípio é uma boa regra para a criação de políticas de acesso. Parece uma regra óbvia mas na prática não é muito fácil de implementar porque é necessário definir as políticas de uma forma extremamente refinada, ou seja, tem que se definir todas as acções possíveis em qualquer recurso para qualquer utilizador, e poderá não ser fácil prever todas as situações.

2.6.6 Autenticação e Autorização

Para um utilizador aceder a determinado recurso é necessário em primeiro lugar que o utilizador se autentique e de seguida seja autorizado a aceder ao recurso pretendido. Um recurso é uma entidade que contém e/ou recebe informação (ficheiro, base de dados, impressora ou outro dispositivo ou aplicação) – Figura 7. Cada um destes processos é explicado mais detalhadamente a seguir.

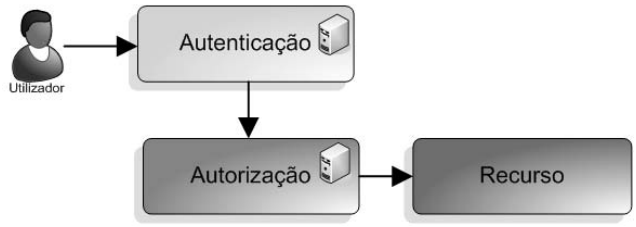


Figura 7 – Passos para acesso a um Recurso

2.6.6.1 Autenticação

O primeiro passo é a Autenticação. O Autenticação é nada mais do que o processo necessário para que uma identidade seja provada. Este é um pré-requisito para o acesso a um determinado recurso. A autenticação tem que responder a duas perguntas, que são elas “Quem és tu?” e “Como é que eu sei que posso confiar em ti?”.

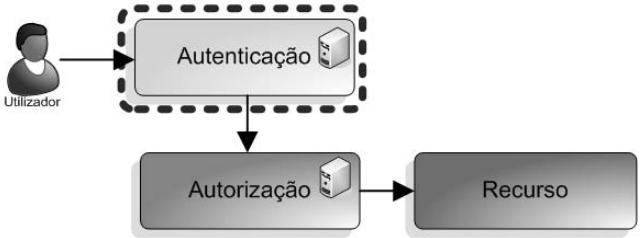


Figura 8 – Passos para o acesso a um Recurso: Autenticação

No mundo físico, podemos responder de variadíssimas formas. Estas formas são credenciais que nos vão dar o direito de aceder a um conjunto de recursos. No mundo digital passa-se da mesma forma. Para ter o acesso a um conjunto de recursos o utilizador tem que apresentar uma credencial.

As credenciais são criadas baseadas em algo que um indivíduo é, faz, sabe, tem, ou uma qualquer combinação destes factores. O nível de segurança aumenta com o número destes factores que são usados. Uma boa regra é quanto mais importante for o recurso a aceder, maior número de factores deve ser utilizado.

Métodos comuns de autenticação são:

- Cookies
- Login e Password
- Desafio de Pergunta/Resposta
- Certificados Digitais
- Dispositivos Biométricos
- *Smart Cards*

A Autenticação é um processo fundamental num sistema de controlo de acessos. A Autenticação verifica que uma entidade pode reivindicar uma determinada identidade e formar uma base para futuras Autorizações e controlo de acessos.

2.6.6.2 Autorização

O segundo passo é a Autorização. A Autorização consiste em verificar quais são os direitos de acesso atribuídos a um utilizador em relação a um determinado recurso.

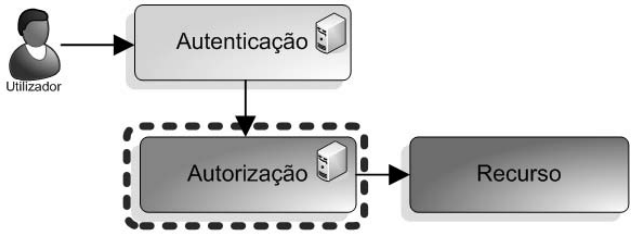


Figura 9 - Passos para o acesso a um Recurso: Autorização

O processo de Autorização tem que garantir que determinados utilizadores têm o acesso a determinado recurso e negar o mesmo a outros utilizadores não autorizados. A autorização implementa políticas de acesso. O objectivo principal é o de preservar e proteger a confidencialidade, integridade e disponibilidade de informação, sistemas e recursos.

2.6.7 Arquitectura Abstracta de um Sistema de Autorização

A arquitectura abstracta de um sistema de Autorização é constituída essencialmente por dois elementos: o *Policy Enforcement Point* (PEP) e o *Policy Decision Point* (PDP). Por exemplo, um utilizador tenta aceder a um ficheiro. A decisão de dar ou não acesso a um recurso é tomada no PDP. A decisão propriamente dita chama-se *Authorization Decision Assertion* (ADA).

Elementos da Arquitectura abstracta de Autorização:

- *Policy Enforcement Point* (PEP) – ponto onde chega o pedido e é executada a política de autorização correspondente.
- *Policy Decision Point* (PDP) – ponto onde é tomada a decisão.
- *Authorization Decision Assertion* (ADA) – é a decisão propriamente dita.

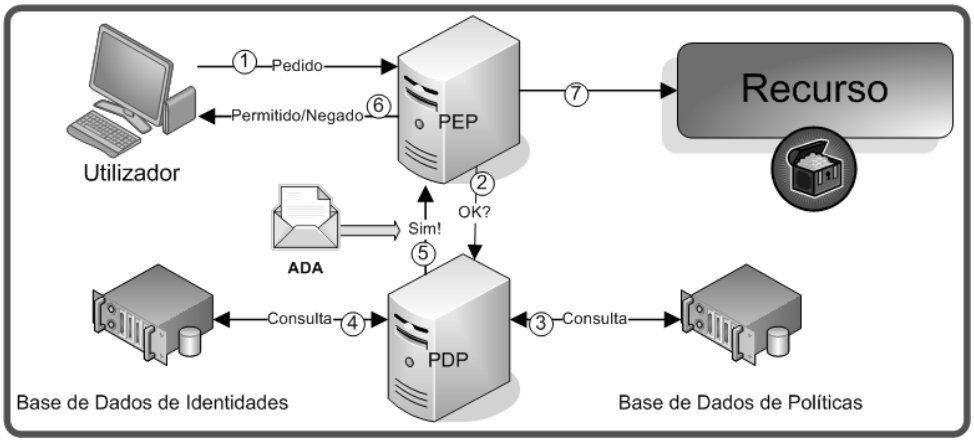


Figura 10 – Arquitectura Abstracta de Autorização

Um utilizador quer aceder a um determinado recurso. Os passos que o sistema de gestão de identidades executa são os seguintes:

1. O utilizador apresenta as suas credenciais.
2. As credenciais são validadas no PEP, podendo ser este um servidor separado.
3. As credenciais são autenticadas usando um qualquer método de prova da identidade.
4. De seguida é consultada a base de dados de políticas de acesso no PDP.
5. O PDP controla os direitos que o utilizador tem sobre o recurso pretendido.
6. O PDP retorna a informação ao PEP através de uma ADA.
7. PEP permite ou não ao utilizador o acesso ao recurso.

2.7 Federação e Login único (SSO – *Single Sign On*)

A proliferação de identidades digitais cria um desafio importante. Os utilizadores têm alguma dificuldade para se lembrarem de múltiplos *usernames* e *passwords*. Cada Web Service ou aplicação requer um *username* e *password*, e são geridos por interfaces de administração separadas.

Uma solução para este problema seria criar uma infra-estrutura central de gestão de identidades, mas esta solução não é escalável. Todos os identificadores locais têm que ser convertidos num identificador único e normalizado, ou então teria de existir uma base de dados para cada um e todos os repositórios. Isto leva à questão de quem vai financiar esta transformação. Um sistema destes seria difícil de manter actualizado e o custo de manutenção seria elevadíssimo. Estando a tecnologia em constante evolução, adicionar uma nova aplicação, função ou até um novo utilizador poderia causar problemas de administração. Além destes problemas, uma solução centralizada seria muito tentador para *hackers*.

Outra aproximação é deixar os diversos recursos da identidade distribuídos pelos seus repositórios originais e criar uma Federação que os ligue entre eles.

Federação de um conjunto de provedores de serviço não é mais do que uma associação a determinada conta de um provedor de identidades. Esta associação consiste no seguinte (Figura 11): o provedor de serviço e o provedor de identidades, com a autorização do utilizador, geram um identificador único – **pseudónimo** – que vão associar às identidades que o utilizador possui em cada um dos domínios. Este identificador único, garante uma maior privacidade ao utilizador, porque torna impossível relacionar a informação acerca de um utilizador distribuída por diversos provedores de serviço. Depois desta federação de contas, quando um dos provedores pretende referenciar o utilizador na comunicação com o outro, é este identificador (pseudónimo) que é usado.

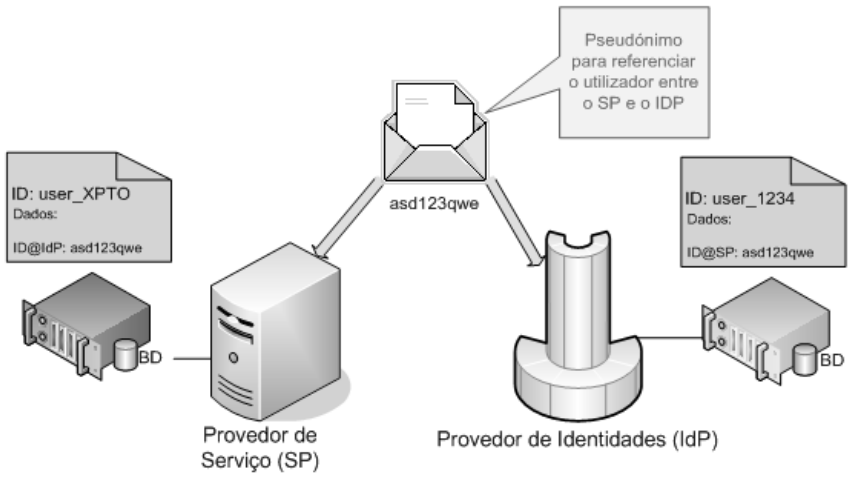


Figura 11 – Federação de Identidades

A federação de identidade possibilita, portanto, aos fornecedores de serviço manter e gerir as suas próprias base de dados de clientes, delegando o processo de autenticação para uma outra entidade (o fornecedor de identidade), assegurando ao mesmo tempo a segurança e privacidade do utilizador e permitindo um mecanismo de **Login único**.

O **Login único (SSO – *Single Sign-On*)** permite que um utilizador apenas necessite de se autenticar uma vez para ter acesso aos diferentes sistemas a que está autorizado a aceder, sem ter que se autenticar em cada um deles.

Este processo de autenticação através de Login único é realizado da seguinte maneira:

1. O utilizador acede a um Provedor de Serviço e procede à sua respectiva autenticação a qual é automaticamente remetida para o Provedor de Identidades ao qual o Provedor de Serviço está associado.

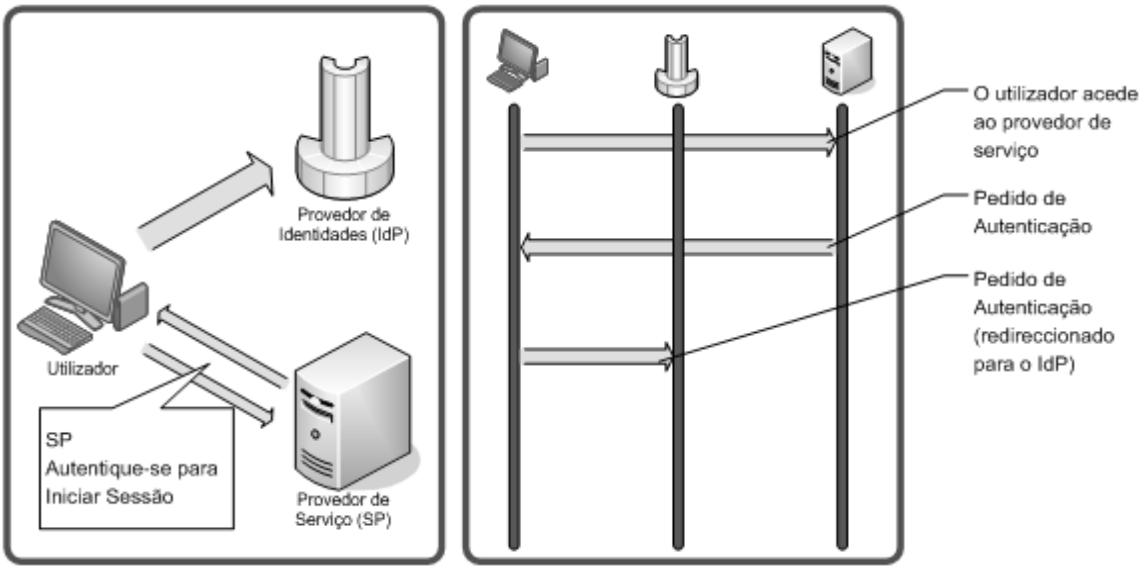


Figura 12 - Autenticação – SSO

2. O Provedor de Identidades verifica se o utilizador já se encontra autenticado. Se não estiver, é mostrada uma janela para autenticação.

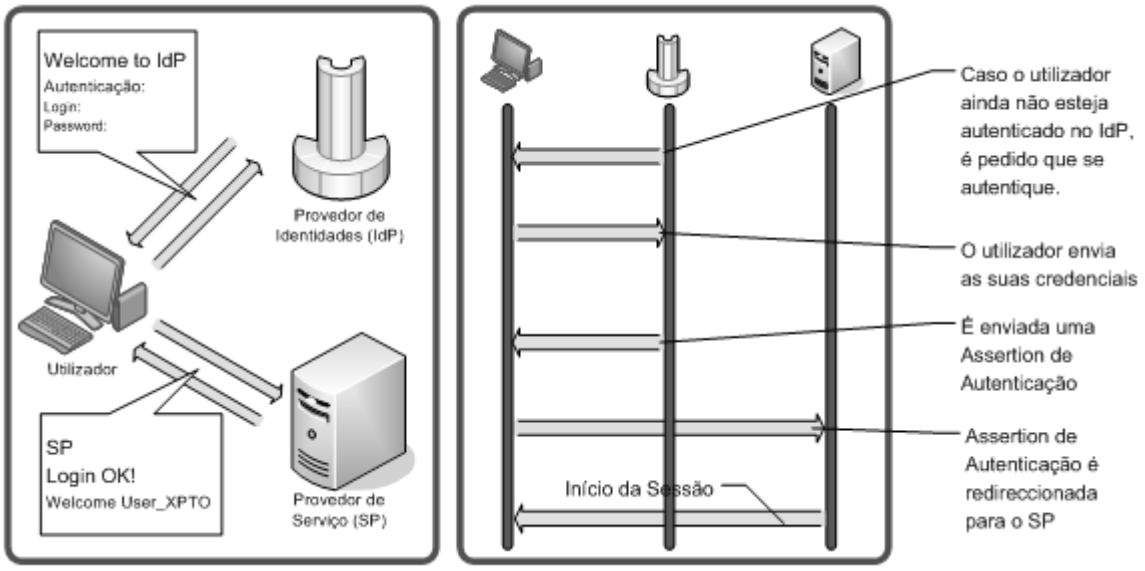


Figura 13 - Autenticação - SSO (cont.)

3. O Provedor de Serviço é informado do sucesso da autenticação.
4. O Provedor de Serviço inicia a prestação de serviços.

Benefícios do *Login* único são a diminuição do desgaste de um utilizador para criar e lembrar das diferentes combinações *username* e *password*, redução do tempo gasto na autenticação para a mesma identidade, reduz o custo de pedidos de assistência técnica por esquecimento de *passwords*.

Neste Capítulo definiu-se Identidade, Identidade Digital, Gestão de Identidades e os conceitos associados. Estas definições são essenciais para o correcto entendimento tanto do Capítulo seguinte que tem como assunto o “Estado da Arte” e de todos os outros Capítulos.

3 Estado da normalização aplicável ao cenário de Gestão de Identidades

Neste Capítulo são abordadas as principais tecnologias e organizações que estão envolvidas no sentido do desenvolvimento e normalização de tecnologias e normas em cenários de Gestão de Identidades.

3.1 OASIS

OASIS (*Organization for the Advancement of Structured Information Standards*) [17] é um consórcio sem fins lucrativos, que conduz o desenvolvimento, convergência e adopção de normas abertas para *E-business* e *Web Services*. O consórcio produz especificações e normas para *Web Services*, para segurança, *e-business* e outros tipos de especificações e normas para mercados de aplicações mais específicas. Fundada em 1993, a OASIS conta com mais de 5000 participantes representando mais de 600 organizações e membros isolados em mais de 100 países. Dentre os membros da organização estão grandes nomes da indústria de tecnologia da informação como IBM, SAP AG e Sun Microsystems.

O consórcio alberga dois portais de informação sobre XML [58] e *Web Services*, que são o *Cover Pages* [18] e XML.org [19].

O consórcio OASIS foi fundado em 1993 sob o nome de SGML *Open* como um consórcio de fabricantes e utilizadores, devotados em desenvolver orientações para a interoperabilidade entre produtos que suportassem *Standard Generalized Markup Language* (SGML) [63]. O OASIS mudou o seu nome em 1998 para reflectir a expansão do trabalho desenvolvido, incluindo o XML e outras normas relacionadas.

Das especificações e normas aprovadas pelo consórcio, existem algumas bastante importantes no âmbito da Gestão de Identidades, que são as seguintes:

- SAML [6][Anexo 1.2],
- XACML [8][Anexo 1.3],
- SPML [10][Anexo 1.4],
- WS-Security [12][Anexo 1.5],
- WS-Trust [15][Anexo 1.6].

3.2 Liberty Alliance Project

A *Liberty Alliance* [20] foi formada em 2001 por aproximadamente 30 organizações para estabelecer uma normas abertas, orientações e boas práticas para a gestão de identidades. Actualmente conta com mais de 150 organizações que participam nas diversas comunidades existentes (SIGs – *Special Interest Groups*).

Implantado por diversas organizações em todo o mundo, a *Liberty Federation* permite a consumidores e utilizadores de serviços Internet e de aplicações de comércio electrónico, autenticarem-se e registarem-se numa rede ou domínio, uma única vez, a partir de qualquer dispositivo e, em seguida, visitar ou utilizar serviços a partir de múltiplos sites. Esta federação não exige ao utilizador que tenha que efectuar uma nova autenticação (*Single Sign-On*) sempre que acede a um serviço que esteja federado, e suporta controlos de privacidade estabelecidos pelo utilizador. Estão disponíveis todos os documentos e estudos, bem como apresentações de implementações. A *Liberty Alliance* cedeu a sua especificação para a federação, ID-FF (*ID Federation Framework*), para a OASIS, formando a base para o SAML 2.0, a especificação que agora a *Liberty Alliance* reconhece.

A *Liberty Alliance* lançou em 2003 a *Liberty Web Services*. Desenvolvida com base em necessidades bem definidas pelas áreas de negócio e com a privacidade do utilizador e consumidor como principal requisito, a *Liberty Web Services* é uma *framework* aberta para a implementação e gestão de uma variedade de *Web Services* baseados em identidades. A *Liberty Web Services* inclui as seguintes aplicações: *Geo-location*, *Contact Book*, *Calendar*, *Mobile Messaging* e *Liberty People Service*, que é a primeira *framework* para um *Web Service* para a gestão de aplicações sociais como *blogs*, *bookmarks*, calendários, partilha de fotografias e *instant messaging* numa rede social federada, segura e respeitando a privacidade de cada um. A *Liberty Alliance* trabalha muito abertamente com outras organizações, adoptando normas publicadas por outras organizações e contribuindo também com trabalho relevante.

A *Liberty Alliance* está também muito centrada sobre nos aspectos dos negócios e políticas de gestão de identidades, publicando para isso orientações em variadas formas para diferentes tipos de negócios e audiências.

3.2.1 Arquitectura da Liberty Alliance

A arquitectura da *Liberty Alliance* é composta por quatro módulos:

- *Identity Federation Framework* (ID-FF)
- *Identity Web Services Framework* (ID-WSF)
- *Identity Services Interfaces Specifications* (ID-SIS)
- Adopção e extensão de normas industriais existentes

A Figura 14 mostra como são organizados estes módulos da arquitectura da *Liberty Alliance*:

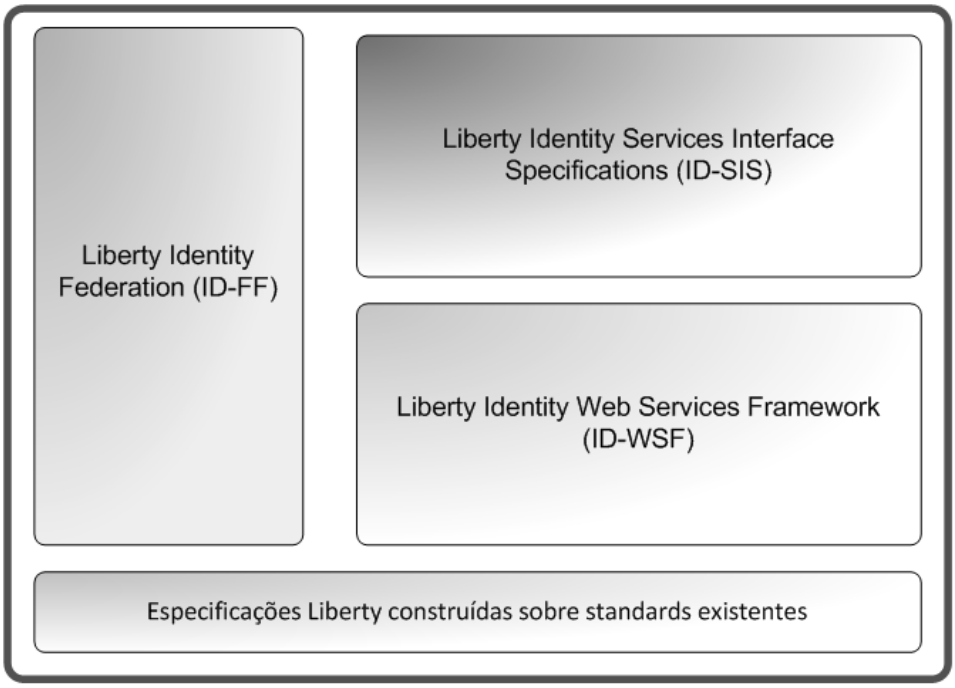


Figura 14 – Arquitectura da Liberty Alliance

3.2.1.1 Identity Federation Framework (ID-FF)

A *Identity Federation Framework* (ID-FF) é o módulo que permite a federação e a gestão da identidade do utilizador. Este módulo tem a possibilidade de ser usado autonomamente ou em conjugação com sistemas de gestão de identidades existentes. Esta infra-estrutura, está concebida de forma a poder operar sobre qualquer tipo de plataformas e com todo o tipo de dispositivos ligados em rede – computadores pessoais, telemóveis e outro tipo de dispositivos que possam vir a ser desenvolvidos no futuro.

Características do ID-FF:

- **Agregação de contas:** permite ao utilizador agregar ou federar as suas diversas contas que suportem as funcionalidades *Liberty* para fins de autenticação.
- **Autenticação Simplificada:** permite ao utilizador fazer uma única vez a autenticação num serviço com funcionalidades *Liberty* e essa autenticação ser válida para todos os outros serviços *Liberty*. A autenticação simplificada é suportada dentro de um *circle of trust*[1].
- **Gestão de Sessão:** permite a entidades ou organizações que agregam contas, comunicarem o tipo de autenticação que deve ser usado pelo utilizador quando este acede ao serviço pretendido. Permite o *Single Sign-off*, que desligada o utilizador de todos os serviços *Liberty* a que tenha acedido quando termina a sessão actual.
- **Anonimato:** permite a um serviço pedir atributos específicos de um utilizador, sem necessitar de conhecer a sua identidade. Por exemplo, para fornecer informação meteorológica personalizada, o serviço meteorológico pode requerer o código postal do utilizador através de um pedido de serviço anónimo, sem tomar conhecimento da sua identidade.
- **Protocolo para o Processo de Descoberta e Troca de Meta data em Tempo Real:** permite que *providers* troquem meta data (esquemas e protocolos a usar) entre si, para que possam estabelecer canais de comunicação entre si. Esta característica do ID-FF possibilita que esta informação seja trocada em tempo real entre entidades que cumprem com as especificações da *Liberty Alliance*.

3.2.1.2 Identity Web Services Framework (ID-WSF)

O ID-WSF é um módulo de suporte que irá apoiar-se na ID-FF. O ID-WSF define uma infra-estrutura que permite a criação, descoberta e utilização de serviços identidade (fornecimento de serviços baseados em atributos de uma identidade). A intenção é permitir às entidades oferecerem aos seus utilizadores serviços personalizados e com mais-valias. O ID-WSF, tal como no caso do ID-FF, faz uso das melhores especificações de segurança disponíveis para oferecer a máxima segurança a entidades e utilizadores.

Algumas características do ID-WSF:

- **ID-WSF *SOAP Binding*:** fornece uma estrutura baseada em SOAP para serviços identidade. Define SOAP *Header blocks* e regras de processamento para permitir a invocação de serviços de identidade via SOAP *request* e SOAP *responses*. Adicionalmente, define um *directive container* para aquelas implementações que desejam usar *rights expression language* já existentes, para especificar políticas do uso de serviços e dados.

[1] Circle of Trust: grupo de serviços e de *Identity Providers* que têm relações de confiança entre si

- **ID-WSF *Security Mechanisms*:** esta especificação define os perfis e requisitos de modo a tornar a descoberta e o uso de serviços identidade seguros. Inclui requisitos de segurança para proteger a privacidade e garantir a integridade e confidencialidade das comunicações entre *service providers* (SP).
- **ID-WSF *Discovery Service*:** de forma a poder providenciar um serviço mais rico ao utilizador, o *service provider* necessita de aceder a partes da informação associada à identidade do utilizador, a qual pode estar espalhada por diversos *providers*. O SP pode usar o serviço de descoberta para determinar a localização de um serviço identidade específico para um utilizador. O serviço de descoberta permite a várias identidades descobrirem de forma dinâmica e segura os serviços identidade de um utilizador, e o serviço responde, com base em permissões, com uma descrição de serviço do serviço identidade desejado.
- **ID-WSF *Data Services Template*:** os *templates* fornecem os blocos para implementar um serviço identidade (por ex. serviço *Personal Profile Identity*) em cima da infra-estrutura ID-WSF. As especificações definem como interrogar e alterar os dados armazenados nos serviços identidades.
- **ID-WSF *Interaction Service*:** um serviço identidade pode ter necessidade de obter a permissão de um utilizador (ou alguém que controla um recurso em nome desse utilizador) para que possa partilhar dados com os serviços que fazem o pedido. A especificação do serviço de interacção, define protocolos e perfis para interacções que permitem a serviços tomarem essas acções.
- **ID-WSF *Profiles for Liberty-enabled User Agents or Devices*:** descreve o perfil e os requerimentos para *Liberty-enabled clients* interagirem com serviços de autenticação.

3.2.1.3 Identity Services Interfaces Specifications (ID-SIS)

O *Liberty Identity Services Interface Specificatons* (ID-SIS) é uma colecção de especificações para serviços inter-operáveis construídos em cima da infra-estrutura ID-WSF. Podem incluir serviços de registo, livro de contactos, calendário, geo-localização, presença ou avisos. Estes serviços independentes serão inter-operáveis através da implementação dos protocolos *Liberty* para cada tipo de serviço. As especificações serão definidas para que seja possível às organizações estenderem de forma rápida e fácil, serviços identidade já existentes, ou criar novos serviços que façam uso da infra-estrutura ID-WSF. Também será possível que outras organizações normativas, trabalhando em conjunto com a *Liberty Alliance*, definam especificações para serviços identidade.

O primeiro ID-SIS a ser disponibilizado é o *Personal Profile Identity Service* (ID-*Personal Profile*). Este serviço define esquemas para a informação básica de perfil de um utilizador. Esta informação consiste normalmente no nome, identidade legal, endereços do domicílio e do emprego e pode também incluir números de telefone, endereços e-mail e alguma informação demográfica, além de detalhes das chaves públicas e outra informação de contacto on-line. Ao fornecer a organizações um conjunto padrão de campos de atributos e valores expectáveis, terão um dicionário ou uma língua comum para comunicarem entre si e assim poderem oferecer serviços inter-operáveis.

3.2.1.4 Adopção e extensão de normas industriais existentes

É neste módulo que estão especificadas as normas já existentes sobre o qual a *Liberty Alliance* desenvolve o seu trabalho.

Estas normas são, por exemplo, as normas especificadas pela OASIS, *World Wide Web Consortium* (W3C) e *Internet Engineering Task Force* (IETF) no que se refere às normas a utilizar: SAML, *WS-Security*, HTTP, WSDL, XML, SOAP, XML-ENC, XML-SIG, SSL/TLS e WAP.

Uma descrição mais detalhada pode ser encontrada em [21].

3.3 Shibboleth

O *Shibboleth System* [64] é uma norma baseada em software aberto para *Web Sigle Sign-On* através de organizações e dentro destas. Permite a *Web Sites* tomarem uma decisão sobre a autorização de acesso a recursos *online* protegidos e preservando a privacidade.

O *Shibboleth* implementa normas para a federação amplamente usadas, principalmente OASIS [17] *Security Assertion Markup Language* (SAML [6]), para fornecer federação, *single sign-on* e uma estrutura para troca de atributos. O *Shibboleth* também fornece funcionalidades de privacidade permitindo que o *browser* do utilizador e o seu *home site* (provedor de identidade) controlem os atributos que são disponibilizados a cada aplicação.

O projecto *Shibboleth* teve inicio em 2000 através do grupo de trabalho MACE [65] para identificar os problemas na partilha de recursos entre organizações, usando estas organizações frequentemente métodos de autenticação e autorização bastantes diferentes. O trabalho de desenho de uma arquitectura prolongou-se por um ano antes de se desenvolver qualquer tipo de implementação. O *Shibboleth* 1.0 [66] foi lançado em 1 de Julho de 2003, *Shibboleth* 1.3 [66] em 26 de Agosto de 2005 e por último (até ao momento) foi lançado o *Shibboleth* 2.0 [67] em 19 de Março de 2008.

3.3.1 Arquitectura

O *Shibboleth* é baseado em tecnologia *web* que implementa o método HTTP/POST, artefactos e perfis do SAML, incluindo o Provedor de Identidades e Provedor de Serviço. O *Shibboleth* 2.0 suporta SAML 2.0 e SAML 1.1.

A Figura 15 mostra o diagrama de troca de mensagens do *Shibboleth* de um utilizador que tenta aceder a um recurso alojado num *Web Site* com suporte para o *Shibboleth*, tendo o utilizador que ser autenticado antes de poder aceder a esse recurso. Existe um WAYF *Server* (*Where Are You From Server*) que tem como função apresentar uma lista de Provedores de Identidades suportados por aquele Provedor de Serviços. Por exemplo, várias entidades e organizações que têm, cada uma, o seu método de autenticação e utilizadores, poderão querer dar acesso aos seus recursos a utilizadores externos de outras entidades com as quais estabeleceram acordos que o permitem. O WAYF *server* fornece ao utilizador uma lista de provedores de identidades que têm um acordo para poder aceder aos seus recursos.

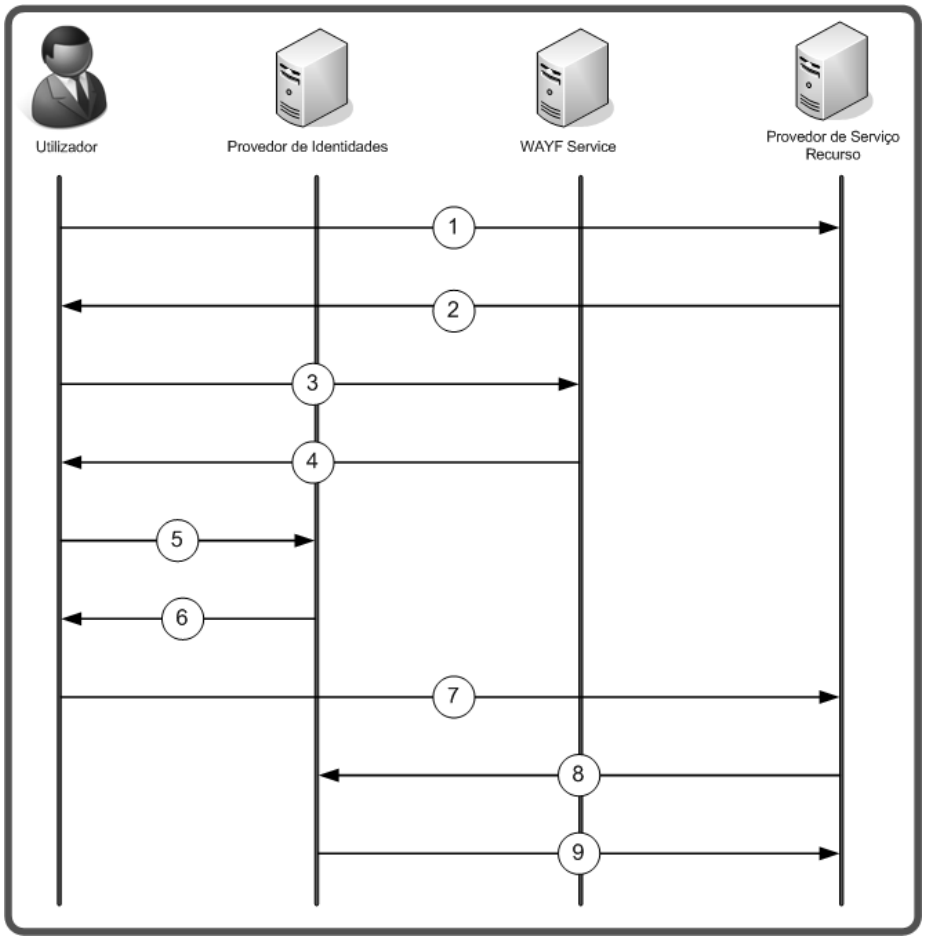


Figura 15 - Diagrama de mensagens da Arquitectura do Shibboleth

1. O Utilizador tenta aceder ao recurso disponibilizado pelo Provedor de Serviço.
2. O Provedor de Serviço redirecciona o Utilizador para o *WAYF Server* porque o utilizador não se encontra autenticado.
3. O Utilizador no WAYF *server* escolhe qual é o seu Provedor de Identidade a partir da lista disponibilizada por esse servidor.
4. O Utilizador é redireccionado para o seu Provedor de Identidade.
5. No Provedor de Identidade é pedido ao utilizador que se autentique.
6. O Utilizador autentica-se, é criado um identificador do utilizador e é redirecionado para o Provedor de Serviço que aloja o recurso que o Utilizador pretende aceder. O *browser* do utilizador recebe um *cookie* para permitir o *single sign-on*.
7. O Provedor de Serviço recebe o identificador.
8. O Provedor de Serviço vai usar o identificador para pedir directamente ao Provedor de Identidades os atributos que necessita para poder tomar uma decisão de autorização.
9. O Provedor de Serviço recebe os atributos e toma então a decisão de permitir ou negar o acesso a esse o recurso ao Utilizador.

3.4 OpenID

O OpenID [22] elimina a necessidade do múltiplos *usernames* e *passwords* nos diferentes *websites* que um utilizador acede.

O utilizador pode escolher o OpenID *Provider* que melhor se adapta às necessidades deste e em que ele confia. O utilizador pode mudar de *Provider* sem que para isso tenha que alterar o seu OpenID. A tecnologia OpenID não é proprietária e é completamente grátis.

O OpenID nasceu da comunidade *open source*. OpenID é uma forma simplificada de identificar indivíduos e que usa a mesma base tecnológica para identificar sites *Web*. Qualquer pessoa pode tornar-se um utilizador do OpenID ou tornar-se num OpenID *Provider* livremente, sem ter que se registar ou ser aprovado por qualquer tipo de organização.

A OpenID *Foundation* foi formada em Junho de 2007 para ajudar a promover, proteger e desenvolver a comunidade e tecnologia OpenID. Tem como missão gerir a propriedade intelectual, marcas e fomentar o crescimento do OpenID. Não tem como missão indicar qual a direcção do desenvolvimento do OpenID, mas sim permitir ajudar e proteger o que é criado pela comunidade.

Algumas organizações membro desta fundação são:

- Facebook
- Google
- IBM
- Microsoft
- PayPal
- VeriSign
- Yahoo!

3.4.1 Arquitectura da troca de mensagens

A Figura 16 mostra quais são as trocas básicas de mensagens quando um utilizador pretende aceder a uma página protegida de um Web Site.

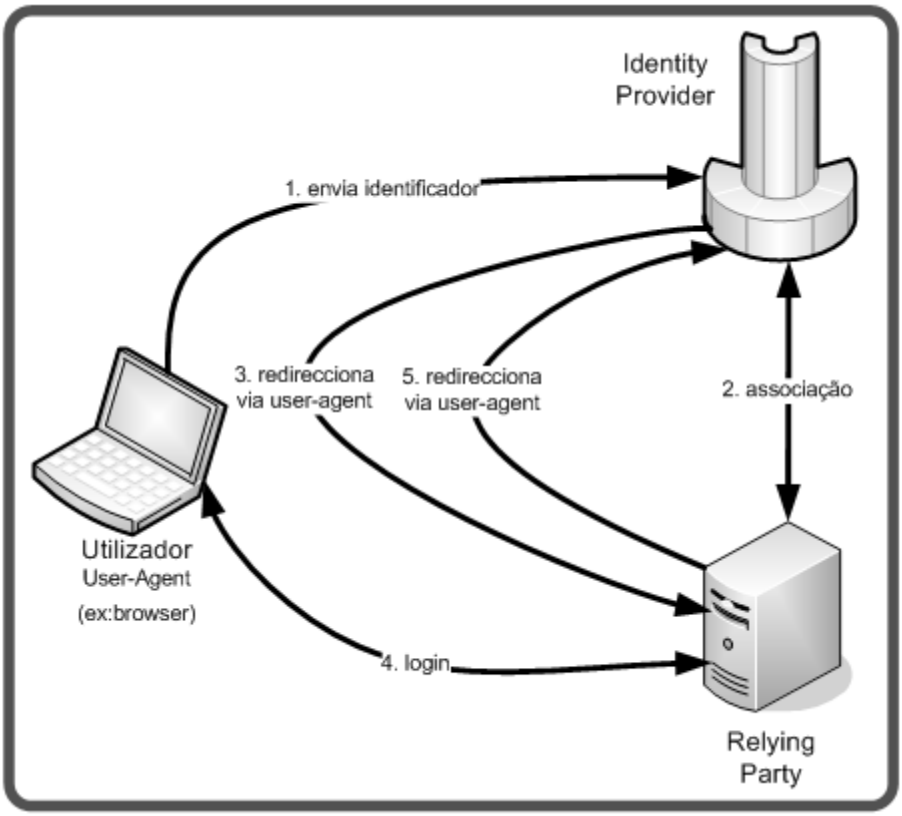


Figura 16 – OpenID em acção

1. O utilizador inicia a autenticação no Web Site (*relying party*) enviando o seu URI (*Uniform Resource Identifier*) usando para isso o seu *User-Agent* (*browser*, …);
   Ex: edseabra.myopenid.com

2. Site normaliza o URI (caso seja necessário);
   O site processa o HTML do OpenID do utilizador;
   Ex: http://edseabra.myopenid.com/
   Encontra o openid.server;
   Estabelece um *shared secret* (segredo compartilhado) com o *Provider*;
3. Redirecciona o *browser* do utilizador para o *Provider*, onde o utilizador se pode efectuar a autenticação;
4. O *Provider* redirecciona o *browser* do utilizador para o site enviando uma resposta;
5. O site verifica a assinatura da resposta e dá acesso ao utilizador.

As mensagens do protocolo de autenticação do OpenID são mapeamentos das chaves (*keys*) de texto sem formatação (*plain-text*) para valores (*values*) de texto sem formação. O exemplo seguinte codifica a seguinte informação:

```
Key     | Value
--------+---------------------------
mode    | error
error   | This is an example message
```

- *Key-Value Form* codificado

```
mode:error
error:This is an example message
```

- x-www-urlencoded, no corpo de um HTTP POST

```
openid.mode=error&openid.error=This%20is%20an%20example%20message
```

O OpenID tem alguns problemas ao nível da segurança. O OpenID é bastante vulnerável a ataques de *Phishing*, *Man-in-Middle Attacks*, entre outros. Se alguém roubar a *password* de acesso do OpenID, esta pessoa terá acesso a todos os sites em que pode usar este método de autenticação. Os problemas de segurança são reconhecidos pelo OpenID, que alerta e tece algumas considerações para minimizar estes problemas de segurança, como por exemplo usar apenas em comunicações seguras, usando SSL com um certificado assinado por Autoridade confiada, o dever dos OpenID *Providers* alertarem e educarem os seus utilizadores para o problema do *phishing*.

Uma descrição mais detalhada pode ser encontrada em [23].

3.5 OAuth

O OAuth [69] é um protocolo aberto para permitir uma API *authorization* segura e normalizada para aplicações desktop, móveis e Internet.

O OAuth teve início por volta dos finais de 2006, quando Blaine Cook estava a trabalhar numa implementação do OpenID para o Twitter. Conjuntamente com Chris Messina, procuravam uma forma de conjugar o uso do OpenID com a Twitter API para delegação da autenticação. Conheceram David Recordin e Larry Halff, entre outros, no encontro CitizenSpace OpenID para discutir as soluções existentes. Depois de analisarem as funcionalidades existentes no OpenID, verificaram que não existia nenhuma norma aberta para uma delegação de acesso API.

Em Abril de 2007, foi criado um Google *group* com um pequeno grupo de implementadores para escreverem uma proposta para um protocolo aberto. Em Julho de 2007 foi criada então um primeiro rascunho da especificação, tendo em Dezembro de 2007 surgido a especificação final OAuth Core 1.0 [70].

O OAuth permite a partilha de recursos privados (vídeos, fotografias, listas de contactos, dados bancários, etc) armazenados num *Web Site* – um Provedor de Serviços – sem ser necessário fornecer o *username* e *password* de acesso a esse *Web Site*. Desta forma cede-se o acesso a recursos privados a terceiros sem ter que existir uma partilha de identidade ou de alguma parte desta.

O OAuth está actualmente implementado em todos os Google Data API's [71]. O OAuth é suportado neste momento por diversas organizações e entidades (Google, Yahoo, Twitter, Flickr, Youtube, Orkut, etc).

3.5.1 Arquitectura da troca de mensagens

Na figura seguinte apresenta-se o diagrama geral de troca de mensagens.

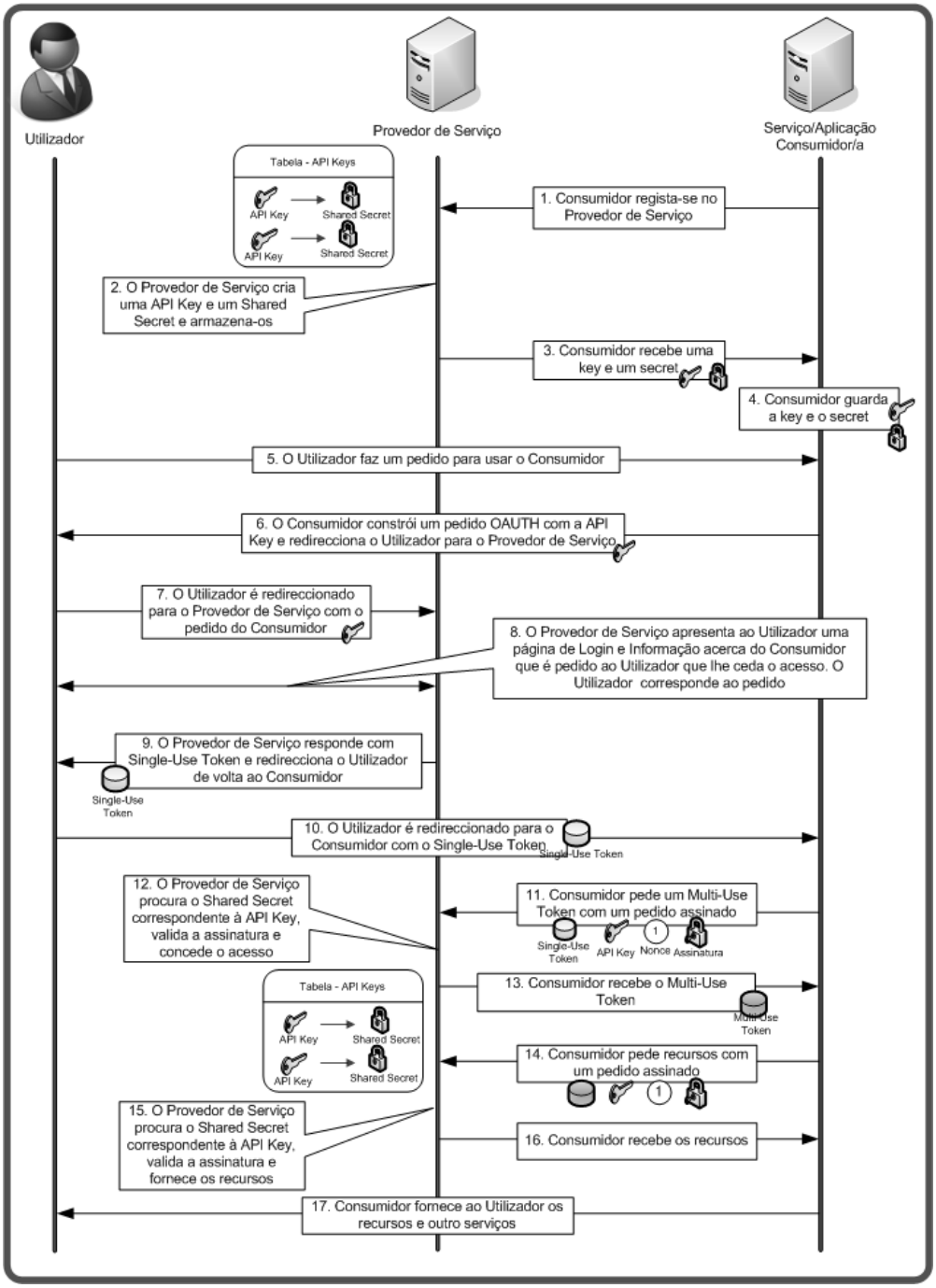


Figura 17 - Diagrama de troca de mensagens da Arquitectura do OAuth

No primeiro passo (1 a 4) o Consumidor regista-se no provedor de serviço que cria um par *key* e *secret* para o consumidor. Ambos guardam esta informação.

No segundo passo (5 a 7) o Utilizador usa pela primeira vez o Consumidor ou pede pela primeira vez para ele ir buscar dados do Provedor de Serviço. O Consumidor constrói um pedido OAUTH para o Utilizador usando a *key* e o *secret* que foram obtidos no primeiro passo e redirecciona o utilizador para o Web Site do Provedor de Serviço com o pedido gerado pelo Consumidor.

No terceiro passo (8 a 10) o Provedor de Serviço pergunta ao Utilizador se o Consumidor pode aceder aos seus dados. As permissões (leitura, escrita) e que tipo de dados o Consumidor pode aceder também são confirmados pelo Utilizador. Depois de ser concedido o acesso do Consumidor aos dados pedidos, o Provedor de Serviço emite um *Single-Use Token* e um *Key pair* que pode ser usado pelo Consumidor para obter mais tarde um *Multi-Use Token* e o *Secret Pair* (também conhecido por *Access Token* e *Secret*).

Um ponto muito importante aqui é que o Provedor de Serviço tem que dizer ao Utilizador que tipo de acesso o Consumidor tem aos seus dados, para que os dados do utilizador não possam ser usados de forma inapropriada.

No quarto passo (11 a 13) o Consumidor pede ao Provedor de Serviço um *Multi-Use Token* usando para isso o *Single-Use Token* e o *Secret Pair* obtidos no passo anterior. Quando o Consumidor faz um pedido de um *Multi-Use Token* usando o *Single-Use Token*, o provedor de serviço valida-o comparando com os valores que tem guardado e verifica a sua assinatura e emite um novo *Token* e *Secret Pair* com tempo de validade, permitindo este múltiplas chamadas ao provedor de serviço. Assim o consumidor pode usar este *Multi-Use Token* para ir buscar dados do utilizador ao provedor de serviço. O *token* e o *secret pair* expiram após um determinado tempo, especificado pelo *Multi-Use Token*. O Utilizador pode usar múltiplos Consumidores para aceder aos seus dados no provedor de serviço, emitindo este um *token* e *secret pair* para cada um dos Consumidores.

No último passo (14 a 17) o Utilizador recebe do Consumidor os dados que pediu.

3.6 Identity Metasystem

O *Identity Metasystem* [25][26] é uma arquitectura inter-operável para a identidade digital, que permite que pessoas tenham e usem uma colecção de identidades digitais baseadas em múltiplas tecnologias, implementações e provedores. Usando esta aproximação, clientes podem continuar a sua infra-estrutura existente, em que investiram, escolher a tecnologia de identidade que funciona melhor para eles, e mais facilmente migrar de tecnologias antigas para novas tecnologias sem sacrificarem a interoperabilidade com os outros. O *Identity Metasystem* é baseado nos princípios descritos em *The Laws of Identity* (As Leis da Identidade) [27].

O *Identity Metasystem* surge com a ideia, talvez a mais provável, que nunca se irá adoptar um único sistema ou tecnologia de identidade digital. Sendo assim, uma solução para esta situação é partir para uma abordagem diferente ao problema, que é a criação de um sistema com a capacidade de ligar a sistemas de identidades existentes e futuros num *Identity Metasystem*. Este meta sistema, sistema de sistemas, poderá proporcionar aos sistemas de identidade que o constituem, ligações mais fortes entre eles, fornecendo interoperabilidade entre eles e permitindo a criação de uma interface comum, consistente e simples para o utilizador. O utilizador sairia bastante beneficiado porque a Internet seria um lugar mais seguro, permitindo aumento substancial de comércio electrónico (*e-commerce*), o combate ao *phishing* e resolver outros problemas relacionados com a identidade digital.

No mundo físico, as pessoas têm inúmeros cartões, de múltiplas formas e feitios nas suas carteiras, como o seu bilhete de identidade, a sua carta de condução, cartões de crédito, etc. As pessoas controlam que cartão querem usar e que quantidade de informação pretendem revelar numa dada situação.

O *Identity Metasystem* permite ao utilizador, de uma forma similar ao mundo físico, manter-se seguro e em controlo quando acede a um recurso na Internet. Permite que o utilizador escolha uma identidade digital da sua colecção e a use num provedor de serviço na Internet. O meta sistema permite que uma identidade digital fornecida por uma determinada tecnologia de sistemas de identidade possa ser usada em sistemas baseados em tecnologias diferentes. Para isto ser possível, existe um intermediário que tem que entender as duas tecnologias, ser confiável e fazer todas as traduções necessárias.

O *Identity Metasystem* não tem como finalidade competir ou substituir os sistemas de identidade que ele liga. O *Identity Metasystem* necessita de sistemas de identidades diferentes para existir.

3.6.1 The Laws of Identity (As Leis da Identidade)

Como foi dito antes o *Identity Metasystem* é baseado nos princípios descritos nas Leis da Identidade [27]. Estas Leis surgiram quando Kim Cameron, um investigador chefe da Microsoft, criou um *blog* em 2004 [24], no qual promovia discussões acerca do tema Gestão de Identidades. O objectivo era perceber o que tinha resultado e o que não tinha resultado, nos actuais e passados esforços para desenvolvimento da Gestão de Identidades, dando mais relevância à compreensão de porque é que se foi num determinado caminho e não se foi por outro. As questões levantadas durante o processo, foram examinadas de múltiplas perspectivas: tecnologicamente, socialmente, usabilidade e privacidade. As diferenças entre fabricantes de software foram postas de parte em nome da compreensão do problema a partir de uma ampla perspectiva da indústria. Nenhum tópico era posto de parte. Um dos tópicos mais estudado foi o das insuficiências do sistema de autenticação universal mais ambicioso na altura, o Microsoft *Passport*. Kim Cameron conseguiu com sucesso que importantes fabricantes, líderes de comunidades se envolvessem em diálogo, obtendo consenso até de proeminentes figuras do *open source* mundial.

Kim Cameron, em 2005, compilou os resultados das discussões num único *white paper*, "*The Laws of Identity*", onde as principais conclusões são resumidas num formato compacto. O *white paper* lista as sete "leis". Elas são princípios que um sistema de gestão de identidades deve cumprir para ser viável. Estas leis tornaram-se imensamente populares e são consideradas por muitos como o manifesto do movimento da gestão de identidades centrada no utilizador. Estas leis definem a arquitectura do *Identity Metasystem*.

As leis são:

1. **O utilizador controla e consente:** Sistemas de identidade só podem revelar informação identificando um utilizador se este consentir.
2. **Divulgação mínima para uso restrito:** A solução que divulga o mínimo de informações de identificação e melhor limita o seu uso é a solução mais estável a longo prazo.
3. **Partes justificáveis:** Sistemas de identidade digital devem ser projectados para que a divulgação das informações de identificação fique limitada às partes que tenham acção necessária e justificável em determinada relação de identidade.
4. **Identidade direccionada:** Um sistema de identidade universal deve ser compatível com identificadores "multidireccionais" (pseudónimos), para uso das entidades públicas, e "unidireccionais" para uso das entidades privadas: facilita a descoberta e, ao mesmo tempo, evita a divulgação desnecessária de dados correlacionados.
5. **Pluralismo de operadores e tecnologias:** Um sistema de identidade universal deve canalizar e activar a interoperabilidade de várias tecnologias de identidade executadas por vários provedores de identidade.
6. **Integração humana:** O meta sistema de identidade universal deve definir o utilizador humano como um componente do sistema distribuído, integrado por meio de mecanismos de comunicação homem-máquina unívocos, que oferecem protecção contra ataques às identidades.

7. **Experiência uniforme entre contextos:** O meta sistema que unifica identidades deve assegurar aos seus utilizadores uma experiência simples e uniforme e, ao mesmo tempo, permitir a separação de contextos por intermédio de vários operadores e tecnologias.

3.6.2 Papeis dentro do Identity Metasystem

Existem três partes participantes, cada uma com o seu papel, dentro do *Identity Metasystem*. Esses papéis são os seguintes:

- **Provedores de Identidade (***Identity Providers***)**, que emitem identidades digitais. Por exemplo, um provedor de cartões de crédito pode emitir uma identidade permitindo um pagamento, empresas podem emitir identidades aos seus clientes, governos podem emitir identidades aos cidadãos, e um indivíduo pode emitir para ele próprio uma identidade (*self-issued*) em contextos como o registo e acesso a uma página na Internet.
- **Provedores de Serviço (***Relying Parties***)**, que requerem identidades. Por exemplo, uma página online ou um serviço online que utiliza identidades emitidas por outras partes.
- **Sujeitos (***Subjects***)**, que são os indivíduos ou entidades acerca dos quais *claims* são feitas. Exemplos são utilizadores, empresas e organizações.

Em muitos casos, as partes participantes no meta sistema desempenham mais que um papel e, muitas vezes, até desempenham os três.

3.6.3 Componentes do Identity Metasystem

Para criar um *Identity Metasystem*, cinco componentes chave são necessários:

1. Uma forma para representar identidades usando *claims*.
2. Meios para provedores de identidade, provedores de serviço e sujeitos poderem negociar.
3. Um protocolo de encapsulamento para poder obter as *claims* e requisitos.
4. Meios para superar as diferenças entre tecnologias e organizações fazendo a tradução, transformação das *claims*.
5. Uma experiência para o utilizador consistente através de múltiplos contextos, tecnologias e operadores.

3.6.4 Identidades baseadas em Claims

Identidades Digitais consistem num conjunto de *claims* feitas acerca do sujeito da identidade, em que "*claims*" são pequenos pedaços de informação acerca do sujeito que o emissor afirma que são válidos. Existe um paralelo com o mundo real. Por exemplo, as *claims* que constam na carta de condução são por exemplo o número da carta de condução, o nome, o endereço, a data de nascimento, a entidade emissora, assinatura, fotografia e os tipos de veículo que o sujeito pode conduzir. A entidade emissora afirma que estas *claims* são válidas. As *claims* num cartão de crédito são por exemplo o nome, o número do cartão, a data de validade, o código de três dígitos de validação, a assinatura e a entidade emissora. A entidade emissora do cartão afirma que aquelas *claims* são válidas. As *claims* de uma identidade emitida pelo próprio sujeito (*self-issued*), onde o provedor de identidade e o sujeito são apenas um e a mesma entidade, podem incluir o nome, o endereço, número de telefone, um endereço de e-mail. Para identidades *self-issued*, o sujeito afirma que estas *claims* são válidas.

3.6.5 Negociação

Negociação permite aos participantes do meta sistema fazer acordos os necessários para eles se poderem conectar entre eles dentro do meta sistema. Negociação é usada para determinar mutuamente tecnologias que ambos suportam, *claims* e outro tipo de requerimentos. Por exemplo, se uma parte entende *claims* SAML e X.509, e a outra parte entende *claims* Kerberos e X.509, as partes vão negociar e decidir que vão usar entre elas *claims* X.509. Outro tipo de negociação determina se as *claims* pedidas por um *relying party* podem ser fornecidas por uma identidade em particular. Ambos os tipos de negociação, são simples exercícios de correspondência. Eles comparam se uma parte pode fornecer com o que a outra parte precisa.

3.6.6 Protocolo de Encapsulamento

O protocolo de encapsulamento fornece uma tecnologia para *claims* e requisitos serem trocados entre sujeitos, *identity providers* e *relying parties*. Os participantes determinam o conteúdo e significado do que é trocado, não o meta sistema. Por exemplo, o protocolo de encapsulamento pode permitir a uma aplicação receber *claims* codificadas em SAML sem ter que perceber ou implementar o protocolo SAML.

3.6.7 Transformadores de Claims

Transformadores de *claims* fazem a ponte entre as diversas organizações e tecnologias traduzindo as claims percebidas num sistema para *claims* percebidas e confiadas por outro sistema. Transformadores de *claims* podem também transformar ou refinar a semântica de uma *claim*. Por exemplo, a *claim* "A nascido a 4 de Julho 1990" pode ser transformada numa *claim* "Mais de 18 anos", que intencionalmente fornece menor informação. Transformadores de claims também podem ser usados para alterar o formato da *claim*. Uma *claim* criada no formato X.509, Kerberos, SAML 1.0, SAML 2.0, pode ser transformada numa *claim* expressa noutra tecnologia. Transformadores de *claims* fornecem a interoperabilidade necessária actualmente e flexibilidade necessária para incorporar novas tecnologias.

3.6.8 Experiência do Utilizador Consistente

Muitos ataques à identidade acontecem porque o utilizador foi enganado por algo que lhe foi apresentado no ecrã, não pela insegurança das tecnologias de comunicação. O ataque "*phishing*" acontece não em ligações seguras entre servidor Web e browser, mas sim entre o browser e o utilizador que o está a usar. O *Identity Metasystem*, tem também como objectivo capacitar o utilizador de fazer decisões de identidade informadas e razoáveis, desenvolvendo para isso uma interface consistente, compreensível e integrada para permitir ao utilizador fazer as escolhas.

Um factor importante para a segurança de todo o sistema será apresentando ao utilizador uma interface fácil de aprender, previsível, que tenha sempre a mesma aparência e funcione da mesma forma independentemente das tecnologias que estão a ser usadas. O utilizador tem que ser informado quais são os itens da sua informação pessoal que o *relying party* está a pedir e para que fim. Isto permite aos utilizadores efectuarem escolhas informadas acerca se devem ou não divulgar essa informação. Finalmente, a interface do utilizador deve fornecer um meio ao utilizador para activamente consentir a divulgação, se ele concordar com as condições.

3.6.9 Arquitectura do Identity Metasystem

A arquitectura do *Identity Metasystem* é formada por um conjunto de especificações ao qual foi chamado WS-* *Web Services Architecture*. Esta arquitectura suporta os requisitos do *Identity Metasystem*.

O protocolo de encapsulamento usado para transformação de *claims* é o *WS-Trust*. Negociações são feitas usando o *WS-MetadataExchange* e o *WS-SecurityPoliy*. Estes protocolos permitem a construção do *Identity Metasystem* tecnologicamente neutro formando a base do próprio. Estes protocolos permitem que novos tipos de identidade e tecnologias sejam incorporados e utilizados à medida que são desenvolvidos e adoptados pela indústria.

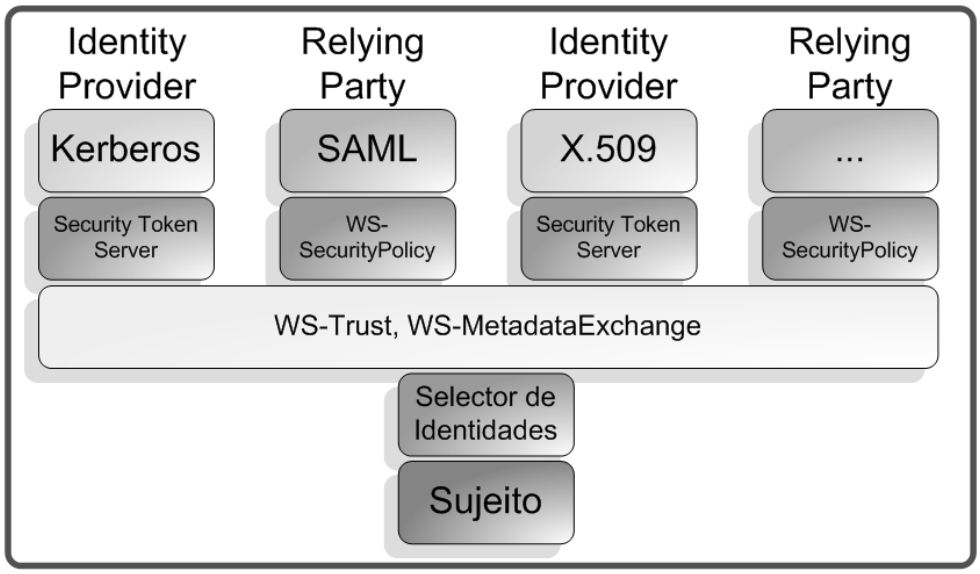


Figura 18 - Diagrama de uma arquitectura do Identity Metasystem

Esta figura apresenta um exemplo das relações entre o sujeito, *identity providers* e *relying parties*, representando algumas das tecnologias usadas pelo meta sistema e pelos sistemas utilizados através do meta sistema.

O *Security Token Server* implementa o protocolo *WS-Trust* e fornece suporte para transformação de *claims*.

Relying Parties fornecem declarações de requisitos, expressos em termos da especificação *WS-SecurityPolicy* e torna-as disponíveis através do protocolo *WS-MetadataExchange*.

Selector de Identidades (*Identity Selector*) implementa uma experiência para o utilizador consistente. Depois de ter sido invocado por uma aplicação, ele realiza a negociação entre o *relying party* e *identity provider*. Mostra ao sujeito (o utilizador) as identidades fornecidas pelos vários *identity providers* que correspondem aos requisitos do *relying party*, obtém as *claims* e disponibiliza-as à aplicação sobre a supervisão do sujeito.

A Figura 19 mostra o esquema de um exemplo de uma transacção canónica de identidade, mostrando quais as normas usadas para implementar cada passo:

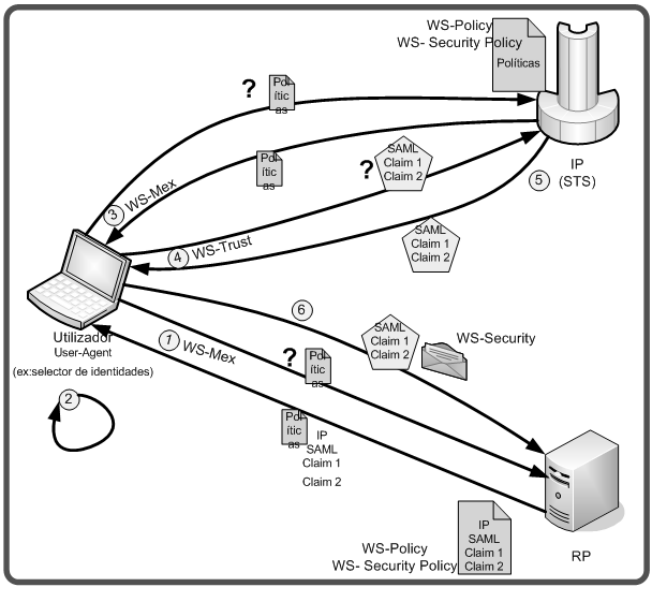


Figura 19 - Esquema de uma transacção canónica de identidade

1. O utilizador pretende invocar o RP. O *user-agent* que o utilizador usa, pede ao RP via *WS-MetadataExchange* quais são as políticas e requerimentos do RP. O RP via *WS-MetadataExchange* devolve um documento *WS-Policy* contendo algumas afirmações *WS-SecurityPolicy*. O RP indica que apenas vai considerar para efeitos de autenticação apenas utilizadores que apresentem uma entidade emitida pelo IP STS, no formato SAML1.1 e contendo duas *claims*, a *Claim* 1 e a *Claim* 2.
2. O *user-agent* do utilizador verifica se o utilizador tem alguma relação com IP que lhe permita pedir ao IP por um *token* com aquele formato e com as *claims* pedidas. De seguida mostra ao utilizador quais são as suas opções, sendo elas, todos os conjuntos de acções que levarão à aquisição do *token* satisfazendo as políticas do RP.
3. Assumindo que o utilizador tem uma relação apropriada com IP, então o utilizador escolhe uma opção que foi oferecida pelo seu agente, e este usa o *WS-MetadaExchange* para perguntar ao IP quais são as suas políticas.
4. O *user-agent* do utilizador usa a informação recebida no passo anterior para pedir uma identidade ao IP STS, enviando-lhe para isso um *Request for Security Token* (RST). O agente também tem o cuidado de segurar o RST usando o tipo de *token* que o IP STS exigiu.
5. O *user-agent* do utilizador recebe o RSTR do IP e recebe o *token* pretendido, e entrega-o o ao utilizador. O utilizador examina o conteúdo das *Claim* 1 e *Claim* 2 e decide se quer ou não divulgar essa informação ao RP.
6. Se o utilizador decidir divulgar a informação ao RP, ele usa o *WS-Security* para segurar o *token* obtido do IP ao RP.

Uma descrição mais detalhada pode ser encontrada em [28].

3.7 Information Card

Como já visto, o *Identity Metasystem* permite aos utilizadores gerir as suas identidades digitais, sendo elas self-issued ou emitidas por um provedor de identidades, e utiliza-las num contexto onde são aceites para obter o acesso a serviços online. No *Identity Metasystem*, as identidades são apresentadas ao utilizador no formato de cartões, os *Information Cards* (*InfoCards*) [29][30]. Os *Information Cards* fazendo uma pequena analogia, não são mais do que versões digitais dos cartões que as pessoas trazem nas suas carteiras.

Figura 20 – Analogia entre os infocards e os cartões que trazemos nas carteiras

A autenticação em *Web Sites* utilizando *information cards* tem como pressupostos os seguintes objectivos:

- Ser independente do *browser* usado
- Ser independente do *Web server* usado
- Ter um impacto mínimo nos *Web Sites*
- Ter uma integração no *browser* harmoniosa
- Provocar uma experiência satisfatória ao utilizador
- Trabalhar com definições de segurança elevadas

3.7.1 Tipos de Information Cards

Como explicado anteriormente uma identidade digital pode ser emitida por um provedor de identidade com Autoridade para o efeito ou pode ser emitida pelo próprio utilizador (*self-issued*). Sendo assim, como os *information cards* representam identidades digitais, existem também dois tipos de cartões:

Self-issued Cards ou *Personal Cards* [31]: são os cartões criados pelo utilizador, agindo este como *identity provider*. O conjunto de *claims* que este tipo de cartões suporta é fixo de forma a que *Relying Parties* possam aceitar este tipo de cartões se assim o desejarem. (No Anexo 2 pode ser consultado o conjunto de claims que é suportado por este tipo de cartões.)

Managed Cards [32]: são cartões que uma Autoridade entrega a um utilizador, e este os importa para o seu selector de cartões. A informação contida neste cartão são apenas as *claims* que suporta e não o seu conteúdo.

A Figura 21 mostra quais são as trocas básicas de mensagens quando um utilizador pretende aceder a uma página protegida de um Web Site.

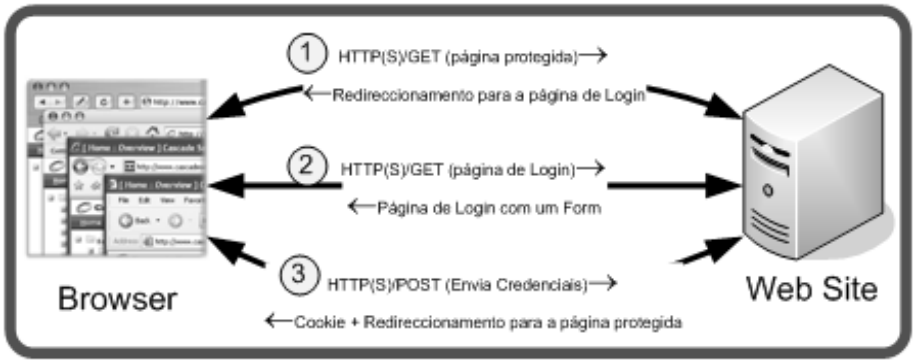


Figura 21 – Cenário comum de Autenticação

Descrição do cenário comum de autenticação:

1. O utilizador acede a uma página protegida que requer autenticação.
2. O site redirecciona o browser para uma página de autenticação que contém um *web form*.
3. O browser envia um *Post* que incluí as credenciais de acesso inseridas pelo utilizador e preenche o *form*. O site valida o conteúdo do *form* que tem as credenciais de acesso do utilizador, tipicamente cria um cookie para o domínio protegido e entrega-a ao browser e

Este é o cenário comum de autenticação num *Web Site*. Um *Web Site* com suporte para *information cards* torna a experiencia de autenticação do utilizador numa experiência diferente. A Figura 22 mostra quais são as trocas básicas de mensagens quando um utilizador pretende aceder a uma página protegida de um *Web Site* com suporte para *information cards*.

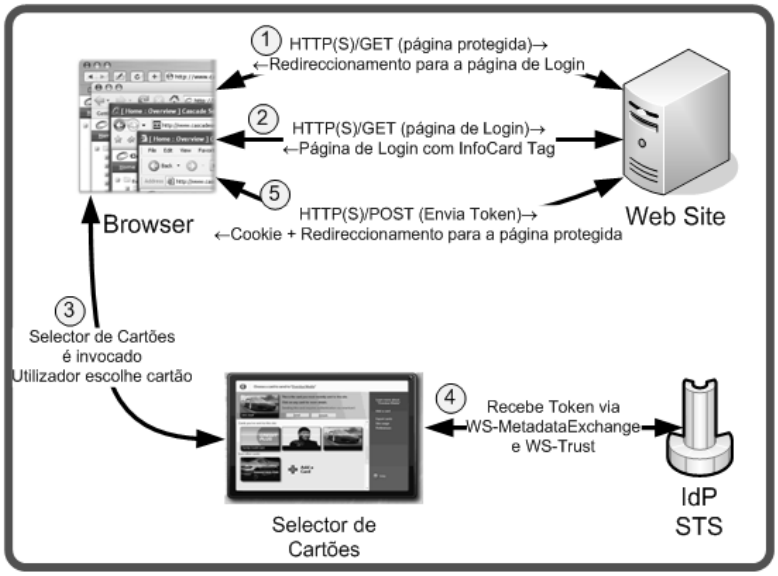


Figura 22 – Cenário de Autenticação usando Infocards

Descrição do cenário fazendo uso de *Information Cards*:

1. O utilizador acede a uma página protegida que requer autenticação.
2. O site redirecciona o browser para uma página de autenticação que contém um *web form* e além disso tem uma HTML *tag* que permite ao utilizador escolher que quer usar Information Cards para efectuar a autenticação. Quando o utilizador selecciona esta *tag*, o *browser* invoca o Selector de Identidades – *Identity Selector*, que implementa, como já visto, a experiência ao utilizador do uso de *Information Cards* e protocolos, e desencadeia o passo (3).
3. São passados os parâmetros requeridos pelo website pela *Information Card* HTML *tag* fornecida pelo *Web Site* no passo (2). De seguida o utilizador usa o selector de identidades para escolher um *information card*, que representa uma identidade digital que pode ser usada para autenticação neste *Web Site*.
4. Neste passo são usados os protocolos normalizados do *Identity Metasystem* já vistos antes (*WS-MetadataExchage* e *WS-Trust*) para ir buscar um *security token*, que representa a identidade digital seleccionada pelo utilizador, ao provedor de identidades para essa identidade.
5. O *browser* faz um *POST* do *token* obtido no *Web Site*, usando HTTPS/POST. O *Web Site* valida o *token* finalizando assim a autenticação do utilizador no *Web Site*. De seguida, o *Web Site*, tipicamente, cria um *cookie* para o domínio protegido e entrega-a ao *browser* e redirecciona-o de volta à página protegida.

3.7.2 Perspectiva do Utilizador

A experiência de utilização de *Information Cards* para o utilizador tem a intenção de ser intuitiva e natural o suficiente para que o utilizador veja esta forma de autenticação como a forma de autenticação. Hoje em dia, *Web Sites* que requerem autenticação, tipicamente, pedem ao utilizador para introduzir um *username* e uma *password* aquando da autenticação. Com *Information Cards*, é pedido ao utilizador para escolher um cartão. Alguns sites poderão aceitar para autenticação apenas *Information Cards* enquanto outros darão a escolha ao utilizador se pretende efectuar a autenticação usando *Information Cards* ou outro tipo de autenticação (*username* e *password*).

3.7.3 Perspectiva do Browser

Para um *browser* suportar *Information Cards* é necessário apenas adicionar pequenas funcionalidades ao *browser*, por exemplo através da instalação de um pequeno *Plug-in*. A principal funcionalidade que tem que ser adicionada é a de o *browser* reconhecer a HTML *tag* especial que invoca o selector de identidades, e passar parâmetros encriptados para o selector de identidades e fazer o POST do *token* resultante da escolha da identidade digital feita pelo utilizador no selector de identidades.

3.7.4 Perspectiva do Web Site

Um Web Site que implemente autenticação com *Information Cards* apenas tem que adicionar na sua página de login a HTML *tag* para pedir o login através de *Information Cards* e adicionar também código para conseguir fazer a validação das credencias recebidas através do *token*. Qualquer outro tipo de código do *Web Site* que não esteja envolvido directamente na autenticação fica exactamente igual, sendo assim simples a implementação da autenticação através de *Information Cards*.

HTML tag

```
<html>
  <head>
    <title>Welcome to MySite</title>
  </head>
  <body>
    <img src='MySite.jpg'/>
    <form name="ctl00" id="ctl00" method="post"
        action="https://www.MySite.com/InfoCard-Browser/Main.aspx">
      <center>
        <img src='infocard.bmp' onClick='ctl00.submit()'/>
        <input type="submit" name="InfoCardSignin" value="Log in"
          id="InfoCardSignin" />
      </center>
      <OBJECT type="application/x-informationCard" name="xmlToken">
        <PARAM Name="tokenType"
Value="urn:oasis:names:tc:SAML:1.0:assertion">
        <PARAM Name="issuer" Value=
            "http://schemas.xmlsoap.org/ws/2005/05/identity/issuer/self">
        <PARAM Name="requiredClaims" Value=
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname">
      </OBJECT>
    </form>
  </body>
</html>
```

Este é um exemplo de uma página que pede ao utilizador para efectuar o *login* usando um *Information Card*. A parte importante desta página é o OBJECT do tipo "aplication/x-informationCard". Quando um cartão é escolhido pelo utilizador, o *security token* resultante é incluído no *POST* como sendo o xmlToken do *form*. Se o utilizador cancelar o pedido de autenticação no selector de identidades, o *POST* resultante contém um xmlToken vazioOs parâmetros definidos dentro do OBJECT, são usados para codificar a informação pedida pelo *WS-SecurityPolicy* em HTML. Neste exemplo, o *relying party* (o *Web Site*) pede um *token* SAML 1.0 de um provedor de identidades *self-issued*, fornecendo as *claims* "*emailaddress*", "*givenname*" e " *surname*". A troca de mensagens é a descrita em cima.

As extensões HTML são usadas para sinalizar o *browser* para este invocar o selector de identidades. Acontece que actualmente nem todos os *browsers* suportam todas as extensões existentes, e mesmo os que suportam, algumas destas extensões são desactivadas quando a configuração do *browser* está definida para uma segurança alta. É o caso da OBJECT *tag* que é comummente suportada, mas que é desactivada quando as definições de segurança são elevadas.

Uma alternativa é o usar uma sintaxe XHTML que não é desactivada quando se alteram as definições de segurança. Contudo, nem todos os *browsers* fornecem suporte total ao XHTML.

Como descrito até agora a HTML *tag* sinaliza o *browser* para este invocar o selector de identidades. Este selector de identidades, como já foi descrito na secção *Identity Metasystem*, permite ao utilizador guardar, gerir e utilizar identidades digitais. Exemplos de selectores de identidade são:

- **Microsoft Windows CardSpace:** que é o software da Microsoft cliente do *Identity Metasystem*. Actualmente é o mais usado visto fazer parte integrante do Microsoft Windows Vista e Microsoft Windows XP SP2.
- **(Bandit Project) DigitalMe:** é um conjunto de componentes que permite aos utilizadores e aplicações interagir com *Web Sites* e serviços compatíveis com *Information Cards* (DigitalMe *Identity Selector*).
- Muitos outros selectores baseados no **Eclipse Higgins Project:** o Higgins é uma *Framework open source* que permite a utilizadores e outros sistemas integrar informação sobre identidade, perfis e relações através de múltiplos protocolos. O Higgins é organizado em três áreas principais: **Higgins Selector**, *Identity Services* e o Higgins *Identity Data Service* (Atributos).

Uma descrição mais detalhada pode ser encontrada em [33].

3.8 OpenID Information Card

O OpenID *Information Card* [34] é uma forma válida de superar alguns dos problemas de segurança referidos anteriormente.

OpenID *Information Cards* podem ser usados para efectuarem autenticações OpenID.

A transferência da emissão das afirmações feitas por um OpenID *Provider*, para um Selector de Identidades torna o OpenID invulnerável ao *phishing*. A forma de funcionamento é em tudo idêntica à explicada acima nos *Information Cards* apenas com a diferença que são OpenID *tokens*. Esta diferença pode ser verificada na *tag* HTML que faz o *browser* invocar o selector de identidades, como se pode ver a seguir:

```
<OBJECT type="application/x-informationCard" name="xmlToken">
 <PARAM Name="tokenType" Value=”http://specs.openid.net/auth/2.0”>
 <PARAM Name="requiredClaims"
        Value=””>

</OBJECT>
```

Uma descrição mais detalhada pode ser encontrada em [35].

Neste Capítulo efectuou-se um estudo sobre as principais tecnologias e organizações que estão envolvidas no sentido do desenvolvimento e normalização de tecnologias e normas em cenários de Gestão de Identidades. Assim, na sua continuação natural apresentam-se no próximo Capítulo cenários de aplicação complementares aos vistos neste Capítulo em cada uma das secções.

4 Cenários de aplicação

Para além dos cenários de aplicação comummente usados na Internet e em outros sistemas e aplicações, e descritos no Capítulo anterior, no âmbito da descrição de cada uma das tecnologias apresentadas, foram idealizados, no âmbito desta Dissertação, um conjunto de cenários de aplicação complementares. Este conjunto de cenários tem como aspectos centrais o papel de um operador de telecomunicações como Provedor de Identidade (*Identity Provider* – IdP)

Qualquer entidade, uma instituição governamental, um operador de telecomunicações ou até inclusive um utilizador pode desempenhar o papel de um Provedor de Identidades. No entanto uma pessoa não sentirá o mesmo grau de confiança em cada uma destas possibilidades. Existirão entidades com mais capacidade e mais experiência que outras, que transmitirão mais segurança ao utilizador e poderão inclusive fornecer outros tipos de serviços e daí trazer vantagens para o utilizador.

A adequabilidade de uma entidade para cumprir o papel de Provedor de Identidades pode ser avaliada tendo em conta alguns aspectos, sendo eles a reputação, oferta de acordos comerciais, oferta de um ponto de acesso e experiência de utilização melhorada [36].

A reputação é um aspecto importante porque, cabendo ao provedor de identidade a responsabilidade de autenticar os utilizadores e de emitir afirmações sobre estes, é natural que um utilizador vá escolher o seu provedor de identidade baseado na reputação da entidade que o vai suportar. O provedor de identidades terá também que divulgar quais são as suas políticas de privacidade, segurança, etc.

A oferta de um vasto número de acordos comerciais entre o provedor de identidade e provedores de serviços, vai dar ao utilizador uma experiência de navegação bastante melhorada e simplificada. Estes acordos comerciais vão permitir, por exemplo, o acesso directo aos serviços sem que o utilizador tenha que se autenticar em cada um deles – *Single Sign-On* – como já visto no capítulo 2. Quanto mais acordos tiver um provedor de identidade, mais vantajoso será para o utilizador.

Idealmente um provedor de identidade deveria ser o primeiro ponto de contacto entre o utilizador e a sua rede de acesso. Quanto mais cedo o utilizador aceder ao provedor de identidade e se autenticar, mais rica poderá ser a sua experiencial *on-line*. Assim o *Single Sign-On* poderá ser usado o mais cedo possível, não havendo, idealmente, lugar a mais autenticações durante a sessão.

Uma boa experiência de utilização é sucesso garantido na satisfação do utilizador. Deverá oferecer uma autenticação simples e intuitiva, suportando um vasto leque de métodos de autenticação para se adaptar às várias situações de utilização do sistema.

Analisando cada um destes aspectos podemos concluir que um operador de telecomunicações está bem posicionado para ser o principal "concorrente" para desempenhar a função de provedor de identidades. Sendo uma entidade em que existe uma relação de confiança entre esta e o utilizador, com os contractos de serviços prestados já existentes, o número variadíssimo de serviços que já são prestados pelas operadoras de telecomunicações e sendo elas o primeiro contacto do utilizador com a rede.

4.1 Cenário 1 – Delegação

A delegação permite a um utilizador delegar, entregar credenciais de acesso a uma entidade, para que essa entidade possa fazer pedidos de serviços em nome do utilizador.

Num cenário de um operador de telecomunicações, delegação será a entrega dos direitos, ou parte, a outros utilizadores ou "instâncias" do mesmo utilizador. Os direitos adquiridos na qualidade de subscritor de um determinado serviço são delegados a alguém, ou algo (pessoa, aplicação ou dispositivo electrónico), mantendo todas as disposições contactuais e financeiras decorrentes da subscrição desse serviço.

Na delegação entram todos os conceitos de políticas de acesso, privacidade, confidencialidade e segurança abordados no Capítulo 2.

Os cenários aqui apresentados são, portanto, baseados no processo de delegar, em que um subscritor de um serviço, que é aquele que é o titular do contracto de subscrição, cria e mantém uma ou várias identidades virtuais, e respectivas políticas de acesso. As identidades virtuais estão associadas à subscrição.

Geralmente, num acesso ADSL ou num serviço de IPTV, o subscritor não é o único a usufruir do serviço. Outros membros da família ou, no caso de uma empresa, variadíssimas pessoas poderão usufruir da mesma instância do serviço. Mas mesmo que o subscritor fosse o único a usufruir do serviço, seria interessante que este se pudesse apresentar perante o serviço com uma identidade diferente, se fosse esse o seu desejo.

Na Figura 23, que representa o exemplo de uma família, temos que uma única subscrição pode representar vários membros da família, e que para além disso cada membro dessa mesma família pode possuir várias identidades, pelas quais se pode apresentar consoante o serviço acedido.

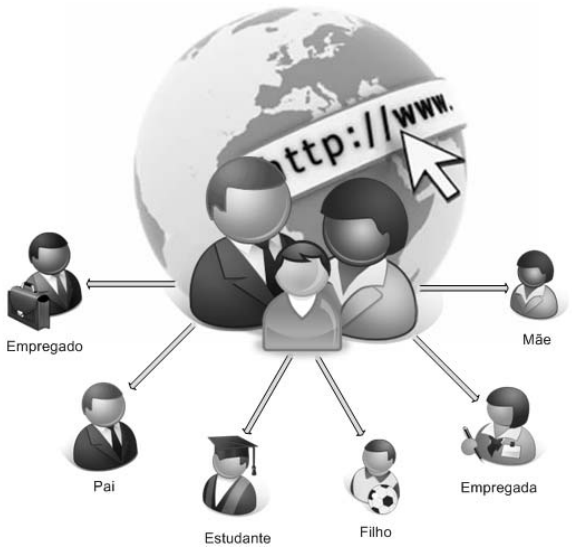


Figura 23 - Identidades associadas a uma subscrição

Figura 24 - Filho apresenta-se como "filho" ou "estudante"

Por exemplo, o filho poderá apresentar-se como “estudante” ou “filho”, dependendo do serviço que este tenta aceder. Se este se apresentar com a identidade “estudante”, poderá apenas aceder, de uma forma automática e sem custos, a um pré-determinado conjunto de recursos, que poderão ser por exemplo a plataforma de *eLearning* da sua escola e outro tipo de recursos didácticos. No caso de se apresentar como “filho”, já lhe será permitido o acesso por exemplo a determinados servidores de jogos mas apenas entre determinadas horas do dia. As possibilidades são inúmeras.

Não é possível fazer a distinção entre quem utiliza o serviço num determinado momento, isto é, para um provedor de acesso ADSL ou de um serviço de IPTV é sempre o titular do contracto, o subscritor, que está a utilizar o serviço. Esta limitação vai impedir uma personalização do serviço, ou a aplicação de regras de acesso consoante quem o está a utilizar.

Para que o processo de delegação seja possível, tem que existir a possibilidade de identificar quem está a utilizar o serviço, ou seja, tem que se conseguir associar uma única subscrição a um conjunto de identidades virtuais e respectivas políticas de acesso e de privacidade. Uma possibilidade consistiria em cada provedor de serviço dar a capacidade ao subscritor de este poder associar à sua subscrição do serviço um conjunto de identidades virtuais e de lhes associar um conjunto de regras e/ou políticas. Mas se este utilizador fosse subscritor de um vasto número de serviços, este teria que criar e manter cada uma das identidades virtuais e políticas associadas em cada um dos serviços, o que rapidamente se tornaria numa situação incomportável ao utilizador ter que fazer toda a gestão dessas identidades e políticas.

A solução para este problema é a utilização de um Provedor de Identidade. Com a utilização de um provedor de identidade, o subscritor tem um único local onde pode criar, modificar, apagar e gerir todas as identidades virtuais e respectivas políticas associadas a cada um dos serviços que este subscreveu. Isto vai implicar que para além do provedor do serviço, o provedor de identidade também tenha que participar na autenticação e autorização do serviço.

A seguir são apresentados dois exemplos onde é demonstrado o uso de delegação como um serviço que uma operadora de telecomunicações pode oferecer aos seus clientes.

Na Figura 25, pretende-se demonstrar as várias relações e associações entre as entidades intervenientes no processo de criação de identidades virtuais e políticas associadas. O utilizador que previamente subscreveu um determinado serviço num provedor de serviço (SP – *Service Provider*), vai federar a sua subscrição com a sua conta no provedor de identidade. Depois de ter efectuado esta federação, já poderá então criar várias identidades virtuais associadas a essa mesma subscrição, assim como as políticas correspondentes (neste exemplo o "Estudante" e o "Filho"). No provedor de identidades, são então criadas as credenciais de acesso ao serviço para as identidades virtuais "Estudante" e "Filho".

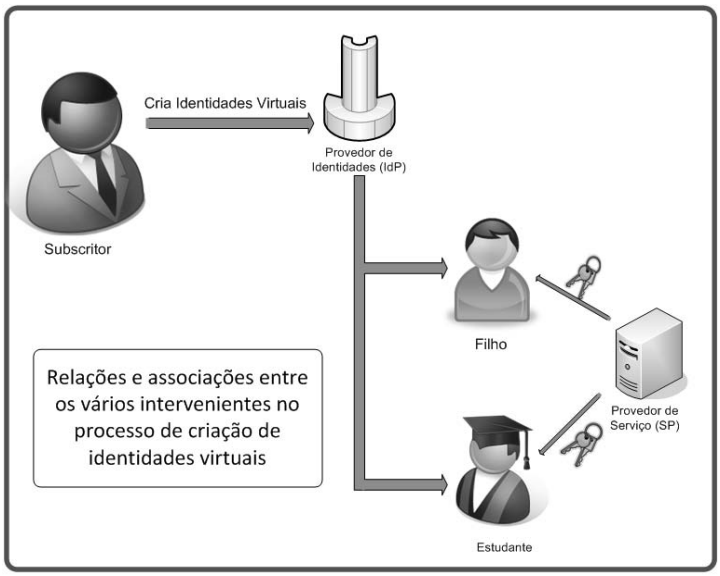


Figura 25 – Relações e associações entre os vários intervenientes no processo de criação de identidades virtuais

Quem tem a responsabilidade do armazenamento das várias identidades virtuais e políticas associadas do subscritor é o IdP. É este elemento que detém, portanto, toda a informação necessária para efectuar a autorização de acesso ao serviço das várias identidades virtuais, consoante o serviço pretendido. Por outro lado, quem tem a função de autenticação da identidade do consumidor do serviço – autenticação da identidade virtual – é o SP. Poderá também, por exemplo, existir uma associação à conta do subscritor para efeitos de taxação. Deste modo, é efectuada uma separação entre os processos de autenticação e de autorização, tanto a nível de execução, como nos elementos que são responsáveis por efectuar esses mesmos procedimentos.

A Figura 26 exemplifica o processo de criação de uma nova identidade virtual. O primeiro passo consiste na autenticação do utilizador perante o IdP, utilizando para isso credenciais que lhe terão sido fornecidas previamente pelo IdP. O utilizador pede a criação de uma nova identidade virtual, indicando quais serão os serviços para os quais essa identidade poderá ser utilizada, associando à nova identidade virtual as políticas de acesso adequadas a cada serviço. O IdP de seguida solicita credenciais para a nova identidade virtual aos SPs, credenciais estas que serão usadas para efectuar a autenticação quando tentar aceder aos serviços em questão. As credenciais são criadas pelos SPs e fornecidas ao utilizador (neste caso o subscritor) através do IdP.

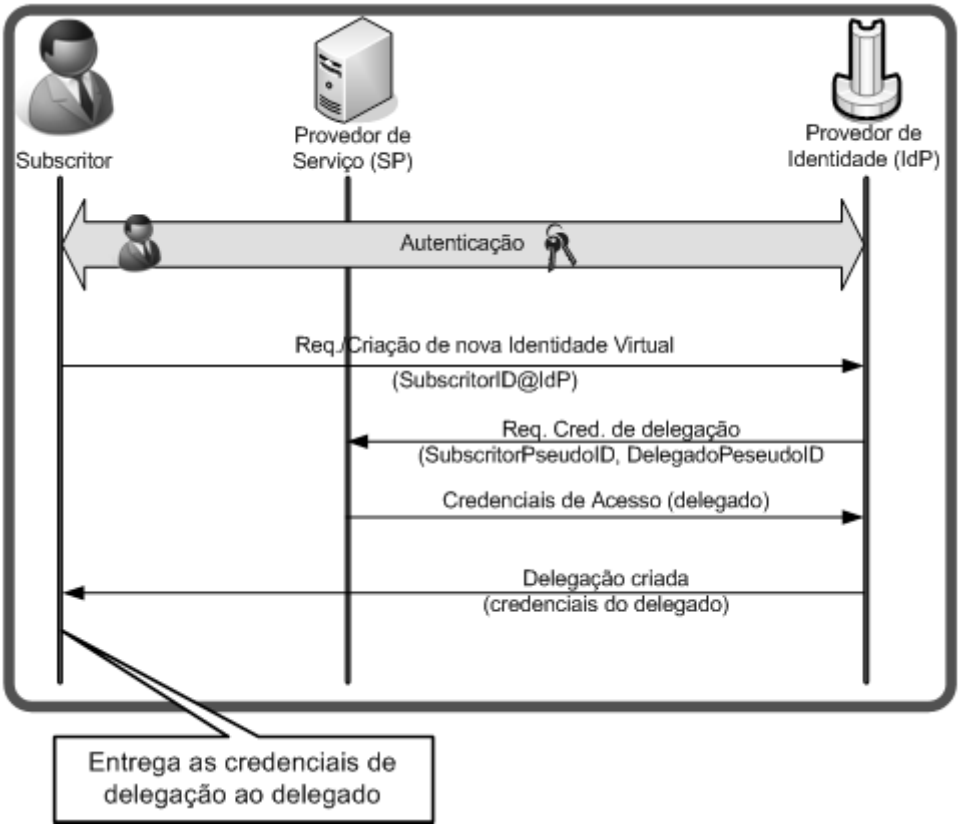


Figura 26 – Criação de uma nova identidade virtual

Na Figura 27 são representadas as várias associações existentes em cada domínio, no domínio do IdP e no domínio do SP. A autenticação de um delegado ("Filho" ou "Estudante" seguindo o exemplo) junto do SP, requer as credenciais para a autenticação, a associação ao subscritor e a informação de qual o IdP que deve contactar para efectuar a autorização desse mesmo delegado, estejam armazenadas no SP. O IdP terá que armazenar toda a informação respeitante às várias identidades virtuais associadas a um subscritor, assim como as respectivas políticas e serviços aos quais devem ser aplicadas, para poder efectuar a autorização de acesso ao serviço por parte de um utilizador.

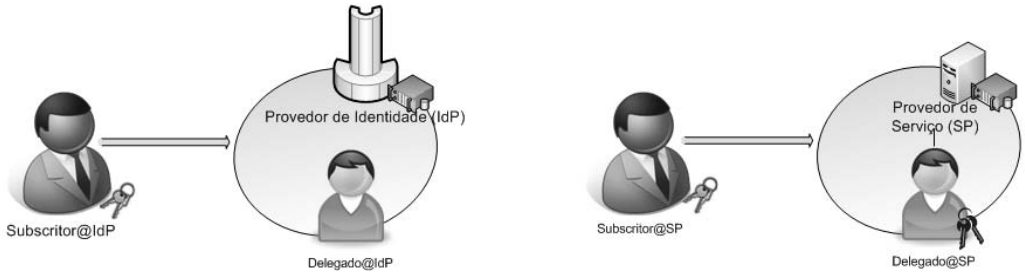


Figura 27 - Domínios e identidades associadas

Um aspecto a salientar é que tantos os identificadores e a credenciais referentes ao subscritor e potenciais delegados, são exclusivos aos respectivos domínios do SP e IdP. O SP necessita de indicar ao IdP a identidade do utilizador que tenta aceder ao serviço para efeitos de autorização e o IdP também necessita de comunicar ao SP qual é o subscritor que este deve associar às credenciais de uma nova identidade virtual para que, por exemplo, o SP possa taxar a conta do subscritor, quando um delegado tenta aceder a um serviço pago. Para que isto seja possível, o IdP e SP têm que estabelecer pseudónimos para associar delegados e subscritores de modo a os poderem referenciar quando necessário. Basicamente, o que se está a fazer é uma federação de identidade com identidades virtuais e com a do subscritor.

Um exemplo de identidades, políticas e identificadores associados ao filho são mostrados na figura seguinte:

Identidade virtual	Políticas	Atributos
"Filho"	Servidor de jogos PERMITIDO apenas das 19h – 20h aos dias de semana	Menor de 18 anos
	ADSL PERMITIDO apenas das 18h – 20h	
"Estudante"	ADSL PERMITIDO das 18h – 20h	Aluno da escola secundária
	Servidor de jogos NÃO PERMITIDO	

Figura 28 - Exemplo de políticas de acesso

4.1.1 Cenário 1 - Aplicado ao IPTV

No cenário aplicado ao IPTV, um delegado, seguindo o exemplo, o “Filho”, tenta o acesso ao serviço de IPTV. Para aceder ao serviço de IPTV, ele vai ter que apresentar as credenciais que o SP disponibilizou aquando da criação da identidade virtual ”Filho” pelo subscritor Figura 29. O SP identifica qual o subscritor associado à identidade “Filho” e qual o IdP que deverá contactar para determinar se o “Filho” pode ou não aceder aos conteúdos pretendidos. O SP para interrogar o IdP vai utilizar o pseudónimo previamente acordado com o IdP para referenciar o delegado “Filho”. O IdP vai verificar quais são as políticas associadas ao “Filho” e ao serviço de IPTV, e de seguida irá confirmar ou negar a autorização de acesso. O SP, mediante a resposta do IdP, irá conceder ou rejeitar o acesso ao conteúdo. Por exemplo, se o conteúdo a que o “Filho” pretendia aceder fossem conteúdos para adultos, o acesso seria negado, pois o “Filho” é menor de 18 anos.

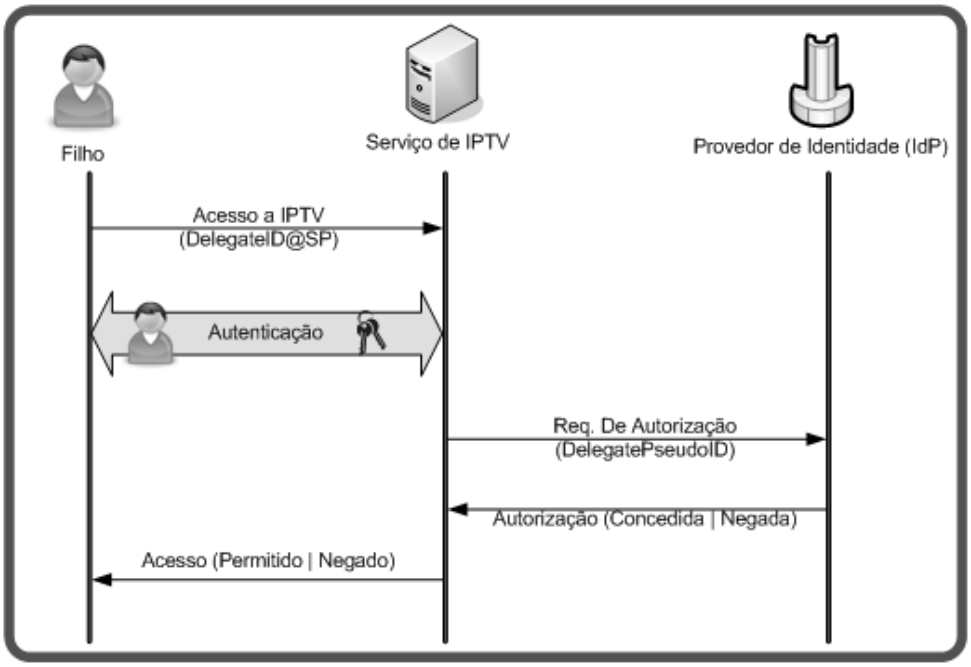


Figura 29 - Acesso ao serviço IPTV

4.1.2 Cenário 1 – Aplicado ao acesso ADSL

No cenário 1 aplicado ao acesso ADSL, o filho inicia o acesso à rede ADSL com a identidade virtual “estudante”. As políticas de acesso definidas no IdP permitem-lhe o acesso ao serviço, logo ele é autorizado. A autenticação do utilizador é feita no provedor de serviço ADSL. De seguida o “estudante” pretende aceder ao serviço de jogos. Quando este apresenta as credenciais de “estudante” junto do servidor de jogos o serviço é rejeitado. O serviço é rejeitado porque o provedor de identidades, com base nas políticas definidas pelo criador da identidade virtual “estudante” para esta, rejeita o acesso ao servidor de jogos.

A identidade virtual “estudante” faz parte dos utilizadores do servidor de *eLearning* da sua escola. O acesso a esta plataforma está integrado com o provedor de identidades. O filho quando se regista na rede com a identidade virtual “estudante”, terá o acesso automático (*single sign-on*) ao servidor de *eLearning* da sua escola. Sendo assim, o utilizador, o filho, não tem que se autenticar no servidor de eLearning da sua escola. O processo reduz-se à autorização dada pelo provedor de identidades ao provedor de serviço *eLearning*.

O diagrama de mensagens seguinte representa o cenário descrito.

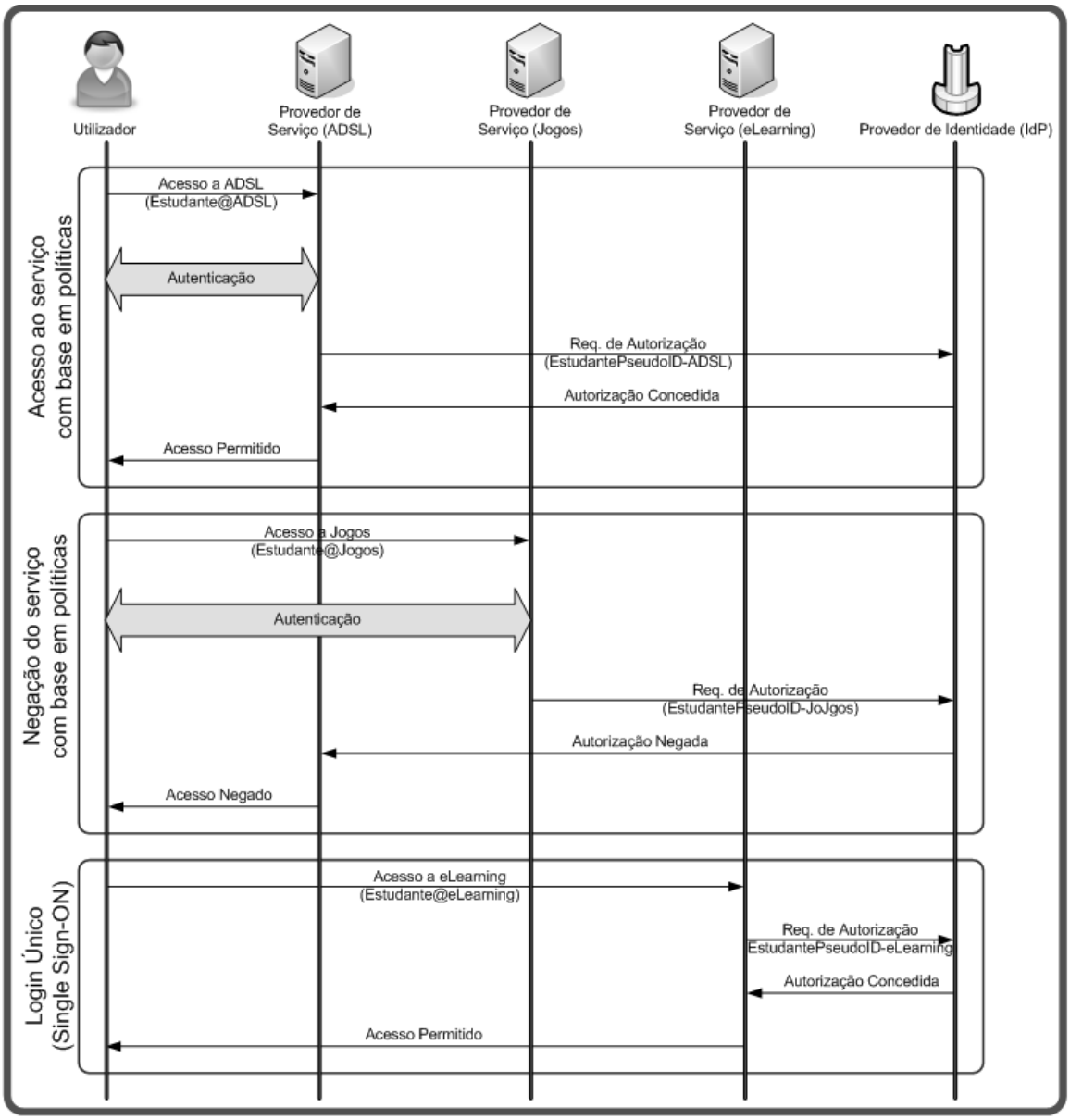


Figura 30 - Acesso a diferentes serviços

4.2 Cenário 2 – Utilização de recursos com base na filiação de grupos

Este cenário é um cenário em que a autorização de acesso a determinados recursos e a gestão de parte da identidade de uma identidade virtual é gerida por uma outra identidade, também esta virtual. O acesso a determinados recursos por parte de um utilizador só é permitido se este acesso for garantido por quem gere os recursos.

Neste cenário apresentado existem duas identidades virtuais: a identidade virtual "médico" e identidade virtual "doente". O doente pode estar em dois locais diferentes, sendo eles o Hospital e Casa. Este cenário tem como objectivo proporcionar ao "doente" quando ele já se encontra em casa, um canal directo de comunicação entre o "doente" e o "médico". Este canal de comunicação é criado e gerido pelo médico. É este quem define quando, como e quais são os doentes que o podem contactar directamente. A privacidade do médico é muito importante e tem que ser levada em conta. O médico não deseja ser incomodado na sua casa ou quando está de férias, no entanto pretende que alguns doentes, quando precisarem, dele o possam fazer. A solução para este problema passa por criar um conjunto de políticas dinâmicas, geridas pelo médico e que se aplicam à identidade virtual "doente".

Quando o doente está no hospital, a sua informação é gerida com base em regras de privacidade criadas pelo doente e existentes num provedor de identidades. O software médico, através do provedor de identidades, associa o "doente" ao "médico". O "médico" pode através do provedor de identidades alterar o perfil do utilizador "doente" e permitir ao "doente" ter contacto com a identidade "médico" quando ambos se encontrarem *online*, mas com base nas políticas de privacidade definidas e mantidas pelo "médico". Assim, o "doente", após sair do hospital, e em sua casa, poderá conversar com o seu "médico", usando um programa de *messaging*, se ambos estiverem online, e as políticas de acesso do "médico" assim o permitirem. Sendo o sistema dinâmico e baseado num conjunto de políticas mantidas pelo "médico", este pode em qualquer altura deixar de permitir que o "doente" tenha a possibilidade de o contactar e conversar com ele.

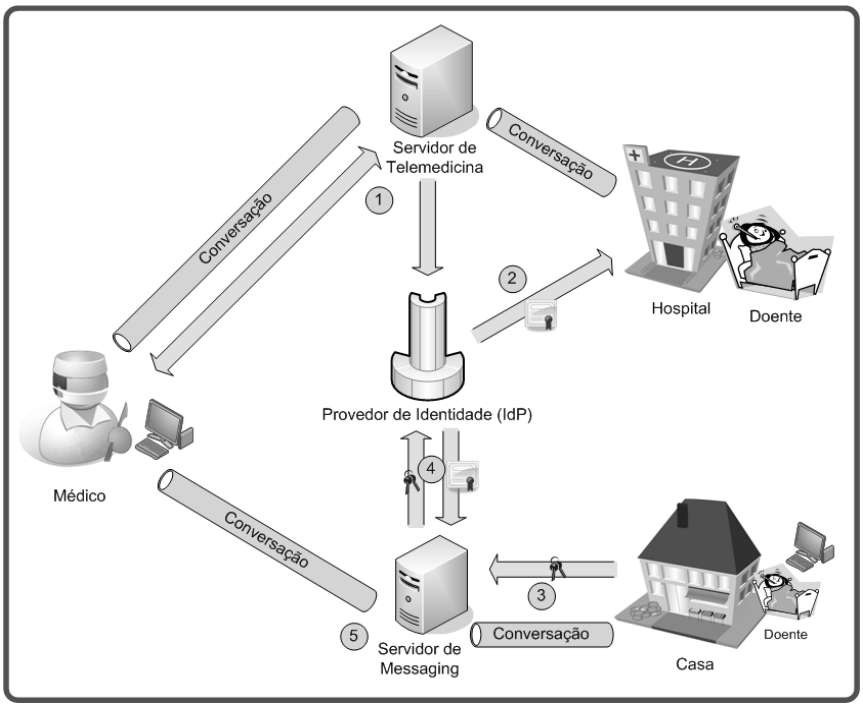


Figura 31 - Cenário 2

A identidade virtual “doente” é um conjunto de identificadores e politicas que dizem respeito ao “doente”, mas parte dela é criada e gerida por uma identidade externa, criando assim um cenário dinâmico onde a privacidade de ambos os actores existe e é mantida por cada um no Provedor de Identidades.

4.3 Cenário 3 – Acesso à rede com base em serviços

Neste cenário, o utilizador não é subscritor do operador de acesso à rede.

Actualmente, para um utilizador aceder a um serviço através de um provedor de acesso à rede, tem que manter um contracto com este provedor de acesso. Para resolver o problema, uma possibilidade é utilizar um provedor de identidades capaz de negociar, em tempo real, com o operador de acesso à rede e com o provedor de serviços, para que o utilizador possa ter acesso ao serviço embora não sendo subscritor do operador de acesso à rede. Neste cenário o processo de autorização e autenticação no operador de acesso à rede é efectuado por uma entidade externa e não pelo próprio utilizador.

A Figura 32 apresenta a descrição do funcionamento de um cenário em que um utilizador não é subscritor do provedor de acesso à rede, mas ainda assim obtém as autorizações necessárias para aceder à rede e a um serviço em particular (acesso ao MyAmazon.com; este provedor de serviço, poderá ter interesse em aceitar o utilizador uma vez que é um bom cliente). O utilizador utiliza um provedor de identidades (IdP) diferente do operador, por exemplo um banco.

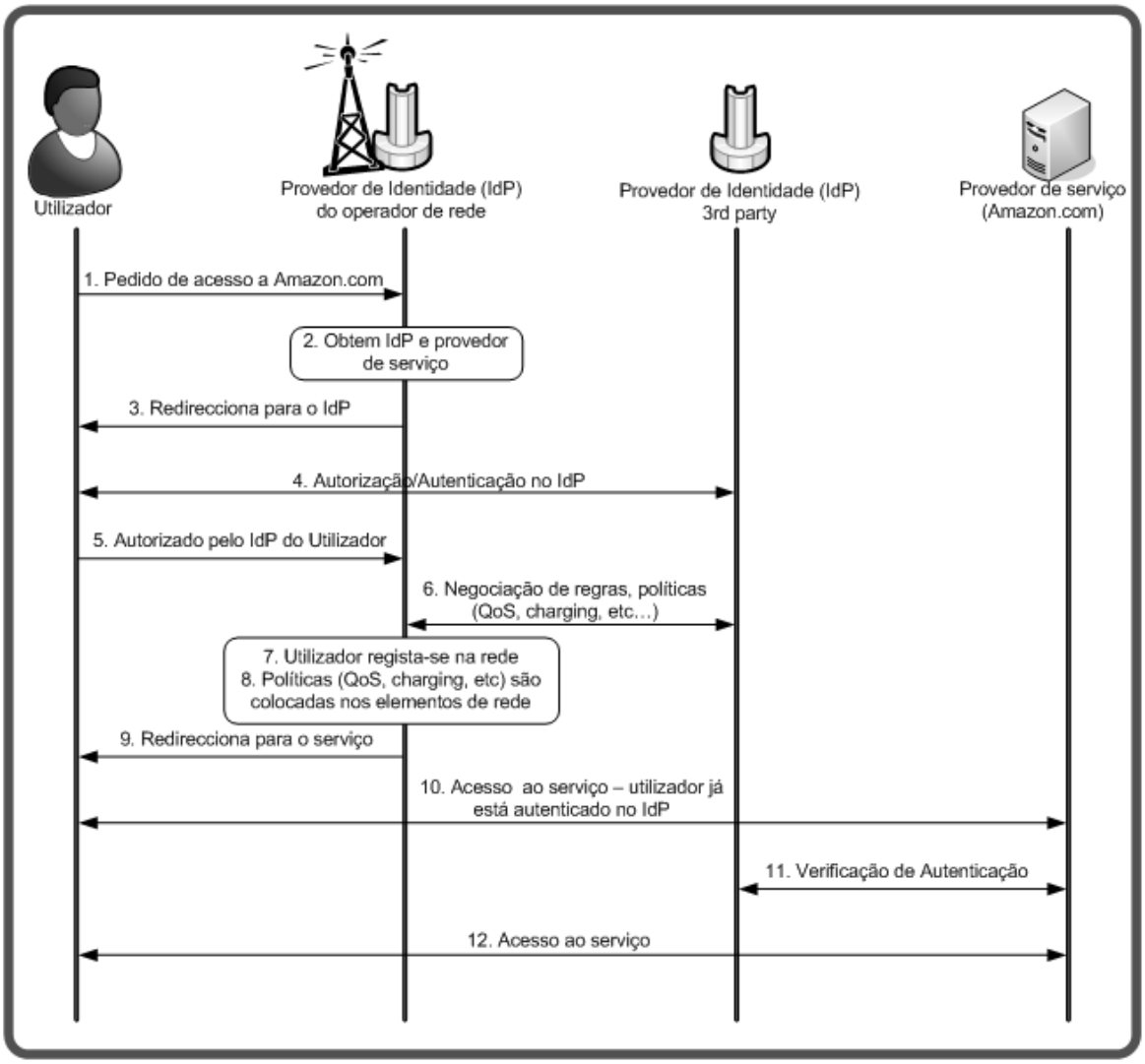


Figura 32 - Cenário de utilização

De seguida é apresentada uma breve descrição do cenário.

1. O utilizador, sem acesso à rede (não é subscritor do operador de rede) tenta aceder ao serviço Amazon.com
2. O operador de rede obtém a informação necessária para redireccionar o utilizador para o *Identity Provider* em que o utilizador existe.
3. O operador redirecciona o utilizador para o provedor de identidades escolhido pelo utilizador
4. O utilizador usa as suas credenciais para se autenticar no IdP

5. O operador de rede é informado que o processo de autenticação do utilizador terminou e que este foi autorizado no IdP
6. Os elementos de gestão de identidades do operador de rede e do provedor de identidades (do utilizador) comunicam para que haja uma troca de informações no que respeita a políticas de acesso, QoS, *charging*, etc... Note-se que o utilizador não é um subscritor do operador de rede, assim este último não tem qualquer informação do utilizador em questão
7. O utilizador é aprovisionado na rede do operador de rede como um "subscritor virtual"
8. O operador de rede aplica as regras negociadas com o provedor de identidades nos elementos de rede
9. O utilizador é redireccionado para o provedor de serviço
10. O utilizador tenta aceder ao serviço, informando-o qual o provedor de identidades onde o utilizador já se encontra autenticado
11. O provedor de serviço verifica se o utilizador já está previamente autenticado com o provedor de identidades
12. O utilizador acede ao serviço.

Embora este cenários seja um pouco diferente dos anteriores, em que o provedor de identidades é a própria operadora, neste caso a operadora poderá ganhar mais-valias uma vez que está aceitar na sua rede um utilizador com o qual não existe um contrato explícito. Este cenário tem algumas semelhanças com o cenário de *Roamming* existente hoje.

As questões relacionadas com a taxação (onde se faz e quem paga o quê?) dependem dos modelos de negócio e poderão existir diversas soluções.

As vantagens do uso de sistemas de gestão de identidades quando aplicados a cenários de um operador de telecomunicações são:

- A autenticação/autorização implícita dos utilizadores facilita o uso dos serviços
- *One-click payments* e micro pagamentos facilitam e estimulam o consumo de serviços de valor acrescentado
- A facilidade de *profiling* por parte do IdP
- A rapidez e facilidade na criação/actualização de dados pessoais (inclusive políticas de privacidade)
- O cliente tem acesso a vários serviços tendo apenas um único ponto de acesso para a gestão dos seus dados, preferências, políticas, etc.
- Serviços disponíveis em multi-plataformas.
- Facilidade na integração de novos serviços.

4.4 Cenário 4 – Acesso a serviço através de autenticação/autorização Out-of-band

Este cenário pretende demonstrar a utilização da Gestão de Identidade em canais de comunicação diferentes dos estabelecidos até ao momento.

Neste cenário uma determinada pessoa pretende subscrever por telefone um determinado serviço, por exemplo um seguro automóvel. Nesta situação é normal e necessário que seja solicitada à pessoa que forneça determinados dados (atributos) como por exemplo a morada, o número de contribuinte, número de bilhete de identidade, e muitos outros. O que acontece é que muitas vezes a pessoa não tem na hora todos estes dados consigo, o que pode levar a que haja alguma dificuldade e demora para subscrever o serviço pretendido.

Numa situação destas, se houvesse alguém que pudesse facultar estes dados, mediante a devida autorização do subscritor, à entidade que está a fornecer o serviço, tornava-se um processo mais simples e rápido. Uma operadora de telecomunicações, é uma entidade que tem todas as condições para o fazer, porque existindo um contracto prévio entre o subscritor e a operadora, é sinal que o subscritor confia na operadora e esta tem os seus dados guardados.

Neste cenário o provedor de serviço, para obter os dados que pretende vai fazer um pedido provedor de identidades da operadora a que corresponde o número de telefone do cliente. A operadora tem que confirmar que este pedido é legítimo e devidamente autorizado pelo subscritor. A operadora envia um SMS para o subscritor que contém um *link* ao qual o subscritor tem que aceder onde lhe é pedida uma confirmação dos dados a serem cedidos.

O subscritor acede via browser do telemóvel à página que identifica quem está a pedir e que dados está a pedir. O subscritor confirma e o provedor de identidades disponibiliza esses dados ao provedor de serviço.

A identidade do provedor de serviço tem que ser confirmada, para que não possam haver pedidos feitos em nome de outros e assim estranhos poderem ter acesso ilegítimo aos dados. Esta confirmação é feita usando certificados. O provedor de serviço terá que ter um certificado emitido por uma Autoridade e assinar os seus pedidos, para que a operadora possa confirmar a sua identidade. Desta forma impede-se assim o uso de técnicas de engenharia social como o *phishing*. O provedor de serviço é obrigado a apresentar o seu certificado electrónico perante o provedor de identidades para comprovar quem diz ser que é.

Na figura seguinte apresenta-se o diagrama de troca de mensagens deste cenário.

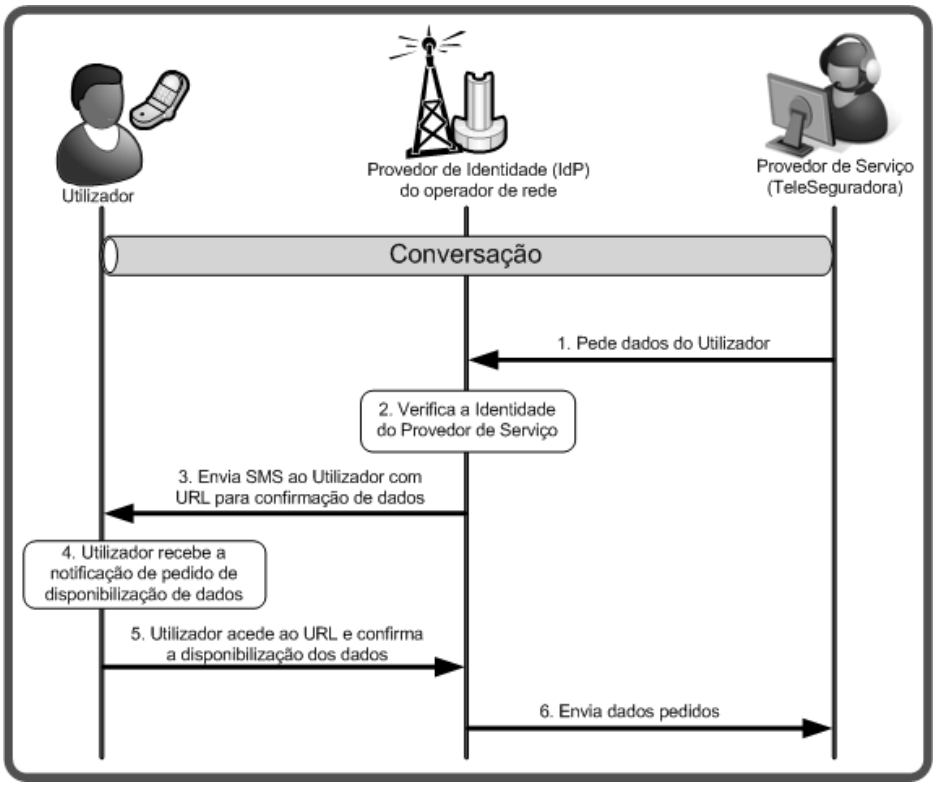


Figura 33 - Cenário de Utilização

Nestes cenários apresentados, o provedor de identidades pode também conter um serviço de taxação, visto que é questionado sempre que um determinado utilizador deseja aceder a um serviço.

Neste Capítulo apresentaram-se cenários de aplicação, dando relevância ao papel que um operador de telecomunicações pode ter neste tipo de cenários.

No Capítulo seguinte apresenta-se a implementação de um protótipo desenvolvido no âmbito desta Dissertação, e a implementação de um cenário de aplicação baseado num cenário apresentado neste Capítulo.

5 Implementação de um Cenário de Aplicação

Um dos objectivos desta Dissertação consistia na implementação de um cenário de aplicação.

Foram testados dois cenários, ambos baseados no *Identity Metasystem* e *Information Cards*. A escolha desta tecnologia teve como base o princípio de que a maior parte dos computadores pessoais actuais têm instalado o *Microsoft Windows XP SP2* ou o *Microsoft Windows Vista*, nos quais o cliente do *Identity Metasystem*, o *Microsoft Windows Cardspace*, é plenamente suportado e no caso do Vista vem instalado por defeito. Sendo assim, a implementação destes cenários não iria exigir do utilizador a instalação de qualquer outro tipo de programa, e adicionando assim todas as vantagens recorrentes do uso de *Information Cards*. Todo o mecanismo de troca de mensagens entre Provedor de Identidades (IdP) e Provedores de Serviços (SP) é o que foi explicado em detalhe no capítulo 3 nas secções 3.6 e 3.7, *Identity Metasystem* e *Information Cards* respectivamente.

5.1 Cenário 1

O primeiro cenário inicialmente proposto para demonstração foi um cenário baseado no "Cenário 2 – Utilização de recursos com base na filiação de grupos" descrito no Capítulo 4. Este é um cenário em que a autorização de acesso a determinados recursos e a gestão de parte da identidade de uma identidade virtual é gerida por uma outra identidade, também esta virtual. O acesso a determinados recursos por parte de um utilizador só é permitido se este acesso for garantido por quem gere os recursos.

O cenário para demonstração era um pouco mais simples que o descrito no Capítulo 4, mas com todos os conceitos a serem demonstrados. O cenário a implementar era o seguinte:

Um doente que esteve internado no hospital e o seu médico, quando lhe dá alta, entrega-lhe por exemplo uma *pen*, ou envia-lhe um e-mail com um *information card* que o doente deve instalar em casa, no seu cliente de *messaging* do seu computador. Este *infocard* vai identificar a identidade virtual "médico", no qual foram definidas políticas de acesso para aquele "doente". O doente apenas tem que ter no seu computador um cliente de *messaging* com suporte para *infocards*, que poderá ser uma realidade num futuro próximo.

Na Figura 34 está representada o diagrama de troca de mensagens do cenário. O médico começa por aceder ao seu IdP, onde vai criar uma identidade virtual “médico” na qual define políticas de acesso para o doente. Efectua o *download* do *infocard* correspondente, e entrega-o ao doente quando este vai para sua casa. O doente por sua vez, instala o *infocard* no seu cliente de *messaging*, indo este consultar ao IdP correspondente quais são as políticas de acesso àquele contacto. Estando tudo como esperado, o contacto é adicionado à sua lista de contactos. O contacto só estará disponível ao doente para efectuar uma conversação, se as políticas de acesso ao contacto assim o permitirem, por exemplo, o médico definiu que aquele doente só o poderia contactar durante a semana das 9h às 18h. Quando o doente tenta iniciar uma conversação, tem que existir uma verificação feita ao IdP correspondente do cartão, para ver se entretanto o médico não revogou aquela identidade virtual. Desta forma, o médico se achar que o doente não necessita de mais assistência, pode impedir que o doente o torne a contactar usando esta via directa de comunicação.

Assim, o doente pode recuperar em sua casa e ter um canal de comunicação com o seu médico, mas só enquanto for necessário, preservando assim a privacidade do médico.

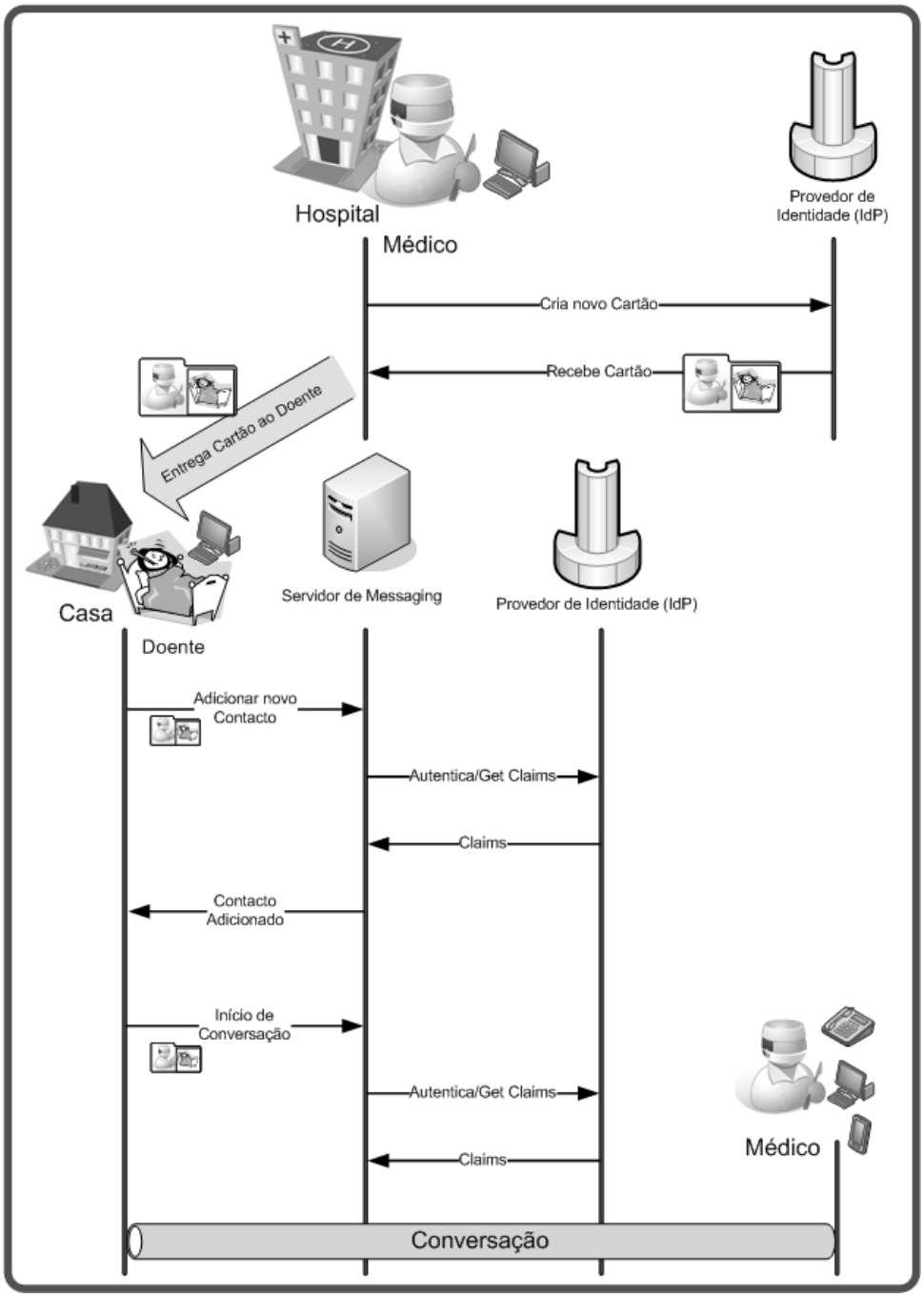


Figura 34 – Demonstração do cenário 1

Para a implementação deste cenário era necessário um IdP onde o médico possa definir quais são as políticas de acesso para o doente e de um cliente de *messaging* com suporte para cartões.

Foi então criado um pequeno cliente *Jabber* com suporte para *infocards* para efeitos de demonstração

O IdP usado teria que suportar *infocards*. O IdP escolhido foi o *WSO2 Identity Solution* [73] - Figura 35.

Figura 35 – WSO2 Identity Solution: Login Page

A escolha incidiu sobre este IdP porque é uma solução pronta a funcionar, está escrito em JAVA [77] e é *open source*. Esta solução não preenchia todos os requisitos, mas era a que se pensou ser mais facilmente e rapidamente adaptável ao cenário em causa e também a outros cenários.

Este servidor de identidades ao nível da gestão de utilizadores e de *infocards* é algo limitado para o que era pretendido demonstrar, aliás, este e todos os outros servidores existentes no mercado. Esta limitação deve-se ao facto de que quando um utilizador cria uma conta num provedor de identidades e lhe são atribuídas credenciais de acesso - Figura 36, são também estas as credenciais que permitem autenticar um *infocard* perante o provedor de identidades. Para este cenário funcionar é necessário que cada *infocard* tenha as suas próprias credenciais para que um doente não possa aceder com as credenciais do seu *infocard* à área de gestão do médico no provedor de identidades.

Figura 36 – WSO2 IS: criação de uma nova identidade virtual sem possibilidade de definir credenciais de autenticação próprias

Devido a esta limitação era necessário modificar o código de modo a permitir então que cada *infocard* tivesse as suas próprias credenciais de autenticação. Tentou-se então alterar o código para satisfazer as necessidades do cenário, mas devido à sua extensão e complexidade resolveu-se desenvolver um pequeno protótipo que fosse facilmente ajustado às necessidades dos mais variados cenários.

5.2 Protótipo My Identity Provider

Partiu-se então para o estudo e desenho da arquitectura do protótipo. O objectivo principal do protótipo era então a criação de um Provedor de Identidades com suporte de *infocards* e com a possibilidade de um utilizador poder criar *infocards* com credenciais de autenticação (*usernames* e *passwords*) distintas, garantindo assim a possibilidade de utilização de *infocards* únicos. As tecnologias escolhidas para a sua implementação foram JAVA [77], JSP [78], SQL [79], Servlet's [80] e Apache Tomcat [81].

Como base deste protótipo usou-se um projecto *open source* muito simples. Este projecto é o XMLdap [68]. O XMLdap é muito basicamente um *Security Token Service* (STS) com uma base de dados muito limitada e suporte de *infocards*.

Depois de um estudo e aprendizagem do código do XMLdap e das tecnologias escolhidas, partiu-se então para a implementação propriamente dita do protótipo. Acrescentaram-se as funcionalidades necessárias para os cenários idealizados, desde a gestão de utilizadores, gestão de *infocards* e a novidade de cada *infocard* poder ter as suas próprias credenciais de autenticação. Nasceu então o *My Identity Provider* v1.0 (MyIdPv1.0).

A seguir mostram-se as figuras com as páginas que compõem o MyIdP v1.0.

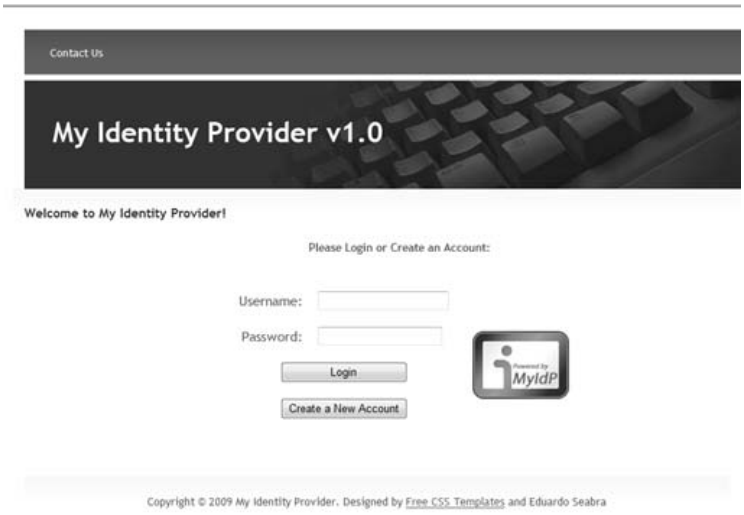


Figura 37 – My Identity Provider v1.0: Página de Login

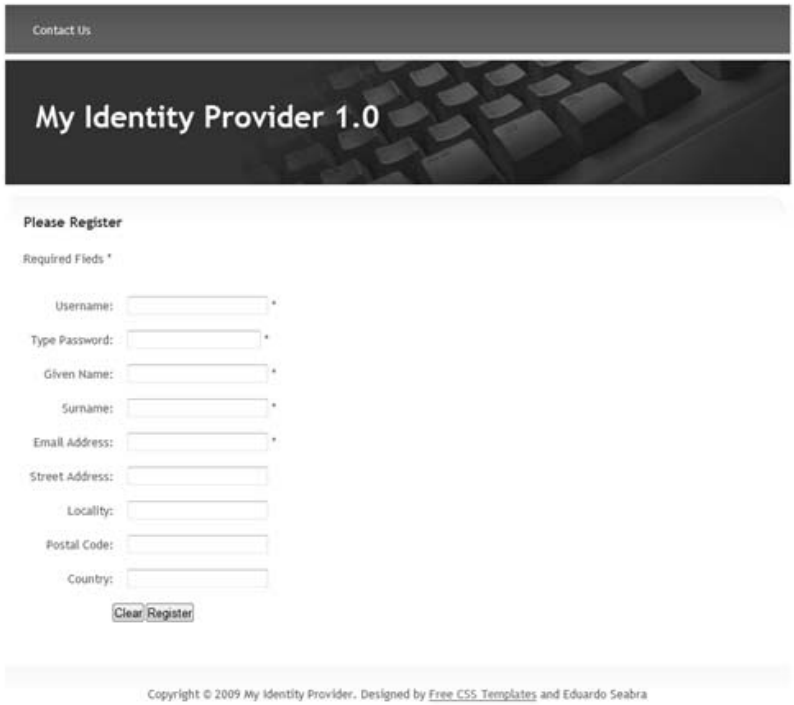


Figura 38 – My Identity Provider v1.0: Página de Registo de um novo utilizador

Na Figura 39, do lado direito pode-se ver o exemplo de dois infocards pertencentes a este utilizador, mas cada um com o seu login e password

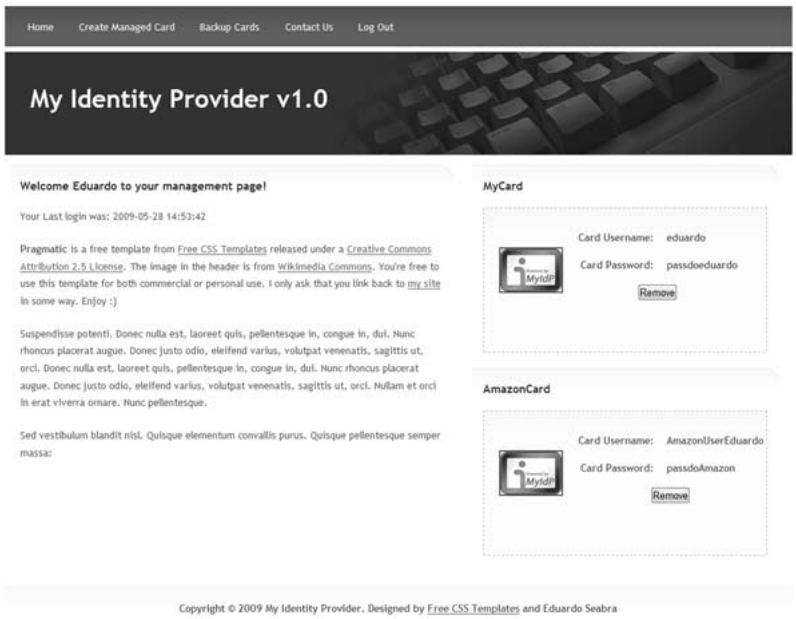


Figura 39 – My Identity Provider v1.0: Página de Gestão da conta do utilizador.

O MyIdPv1.0 é um provedor de identidade totalmente funcional, com suporte para *infocards* com credenciais únicas e facilmente adaptáveis a qualquer cenário.

Na Figura 40, mostra-se a página de para criação de novos infocards, com a novidade de permitir que os infocards tenham credenciais de autenticação diferentes, para permitir que seja entregue a um elemento terceiro.

Figura 40 – My Identity Provider v1.0: Página de para criação de novos infocards.

Se compararmos a página de criação de novos *infocards* do MyIdP v1.0 (Figura 40) com a do WSO2 IS (Figura 36) verificamos que a única diferença é a adição dos campos *Username* e *Password*, mas o que torna o provedor de identidades muito mais versátil. Os campos representados são apenas um exemplo e compõem o conjunto de *claims* dos *self-issued managed cards*, falados no Capítulo 3. Com os *infocards* emitidos pelo MyIdP é possível registar um utilizador em qualquer Web Site que suporte *self-issued managed cards*.

Com este protótipo é possível demonstrar qualquer tipo de cenários, mas achou-se que se poderia ir mais longe. Então surgiu a ideia de criar mais uma inovação no que respeita a provedores de identidades: acrescentar ao MyIdPv1.0 a capacidade de trabalhar com múltiplos cenários em simultâneo, ou seja, que pudesse suportar múltiplos conjuntos de definições de claims em simultâneo. Com isso seria possível, por exemplo, trabalhar com o "Cenário 1 – Delegação" e com o "Cenário 2 – Utilização de recursos com base na filiação de grupos", ambos descritos no Capítulo 4. Depois de analisado todo o código do MyIdPv1.0 concluiu-se que toda a parte de acesso à base de dados teria que ser reescrita para poder suportar qualquer tipo de *managed cards* em simultâneo e parte do *Security Token Service* também teria que ser reescrito, para ser possível identificar, validar, autenticar e carregar dinamicamente as *claims* correspondentes àquele *managed card* quando lhe é solicitado a emissão de um *token* com um determinado conjunto de *claims*. Nasceu assim o *My Identity Provider* v1.5 (MyIdPv1.5).

A página de login e de registo é igual à do MyIdPv1.0 e sendo as diferenças na página de gestão do utilizador, que agora tem uma lista com os serviços que o MyIdP suporta e respectivo botão para a criação do respectivo *managed card*. Depois existe como seria de esperar uma página de criação de *infocards* gerada automaticamente para cada tipo de *infocard* suportado. A seguir apresentam-se algumas figuras (apenas as que são diferentes da versão 1.0) para mostrar a interface do utilizador desta versão.

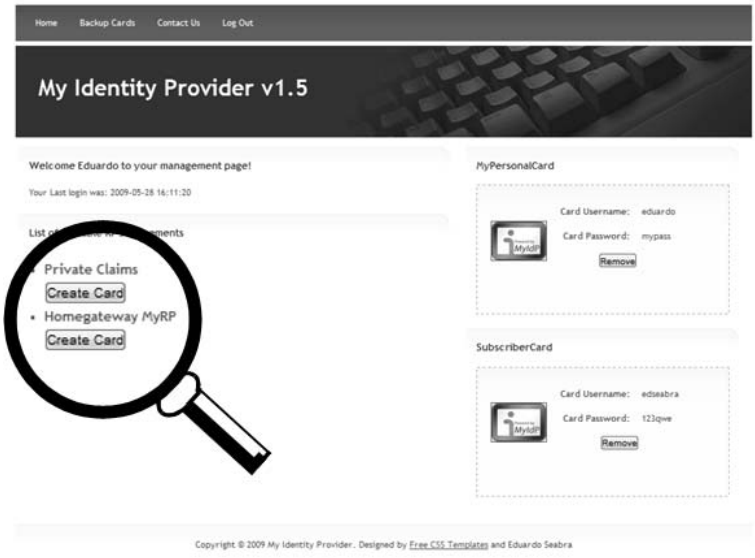


Figura 41 - My Identity Provider v1.5: Página de Gestão da conta do utilizador com a possibilidade de o utilizador criar mais que um tipo de cartões

Figura 42 - My Identity Provider v1.5: Página para criação de self-issued managed cards

Figura 43 - My Identity Provider v1.5: Página para criação de outro tipo de infocards

5.2.1 Arquitectura

O My Identity Provider está dividido em vários blocos funcionais sendo cada um dos blocos constituído por várias classes e servlets, sendo os mais importantes representados na Figura 44.

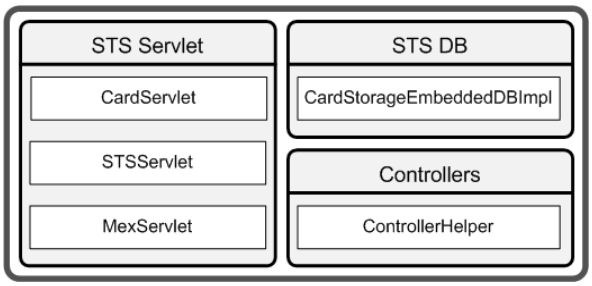


Figura 44 - Blocos Funcionais do MyIdP

O bloco STS Servlet, é o bloco responsável basicamente pela comunicação com o exterior. O CardServlet é a classe responsável pela criação dos Managed Cards. O STSServlet é a classe responsável pela emissão de tokens com os atributos que correspondem a um determinado

Managed Card. O MexServlet é o servlet responsável por indicar quais são as políticas e endereços que devem ser usados (endereço do STSServlet) para pedir atributos.

O STS DB é o bloco responsável por toda a gestão da base de dados de utilizadores e de information cards, implementada na classe CardStorageEmbeddedDBImpl.

O bloco Controller, tem como função gerir toda a interface web de gestão e sessões dos utilizadores, e é implementada pela classe ControllerHelper.

A Figura 45 mostra o diagrama de classes das classes mais importantes. No Anexo 3 pode-se encontrar cada uma destas classes representada em pormenor.

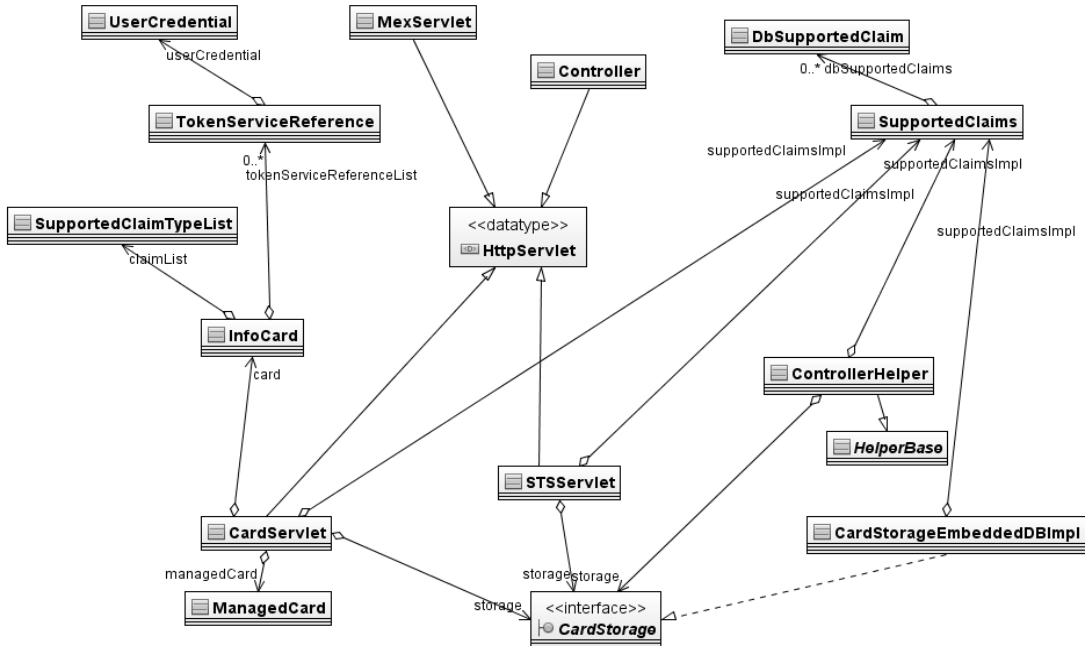


Figura 45 - Diagrama de Classes

As alterações introduzidas nesta versão 1.5 permitem que num futuro desenvolvimento deste protótipo seja possível evoluir no sentido da criação de mais uma novidade neste tipo de provedor de identidades. Será uma versão 2.0 que permitirá para além das inovações da v1.5, a adaptação automática do provedor de identidades a um novo serviço. Por exemplo, se um utilizador quiser usar o provedor de identidades que permite por defeito por exemplo usar o MyIdP num cenário de delegação ADSL e IPTV, e quiser utilizar outro serviço, por exemplo o Amazon.com do cenário 3 do Capítulo 4, basta indicar ao provedor de identidades qual é o novo serviço que pretende usar que o provedor de identidades irá de uma forma autónoma e automática negociar o conjunto de *claims* que terá que suportar para poder suportar o novo serviço. Como será óbvio, o provedor de serviço também terá que suportar esta característica para poder responder ao provedor de identidades. Em termos de negócio, este tipo de funcionalidades poderá criar mais-valias se por exemplo o provedor de identidades do utilizador for o ISP. Assim é com naturalidade e satisfação que o subscritor de determinado ISP vê que pode associar um qualquer serviço ao seu provedor de identidades.

5.3 Cenário 2

Depois do MyIdPv1.5 completamente funcional passou-se então para a implementação de um novo cenário para efectuar uma demonstração. O cenário escolhido foi um cenário baseado no "Cenário 1 (Delegação) – Aplicado ao acesso ADSL" explicado em detalhe no capítulo 4. Existem algumas diferenças, sendo a mais importante, a de a autorização ser feita na *Home Gateway*, na casa do subscritor. Na generalidade das ligações de Internet, existe um único endereço de IP público atribuído ao modem que depois todos os utilizadores dessa ligação vão partilhar entre si. Na impossibilidade actual de atribuir um endereço de IP público a cada um dos utilizadores, a única forma de poder efectuar algum tipo de controlo de acesso actualmente seria na *Home Gateway* de acesso à Internet. Assim mediante políticas previamente definidas pelo subscritor da ligação, políticas de acesso são definidas, por exemplo, horas de acesso, endereços que determinado utilizador pode aceder e em que condição pode aceder. Todo o controlo é feito na *Home Gateway*. Adicionou-se ainda a este cenário uma simples demostração de *Single Sign-On* (SSO) usando cookies. É uma solução que apenas permite demonstrar o SSO dentro de um determinado domínio, por exemplo o domínio do operador de telecomunicações, permitindo assim que o utilizador seja conhecido nos serviços oferecidos pelo operador.

Como referido no início deste capítulo este cenário baseia-se na arquitectura do *Identity Metasystem* e *Information Cards*. Usou-se também o protótipo MyIdP v1.5, uma gateway com um sistema operativo Linux *embedded* no qual se instalou um servidor Apache Tomcat para suportar o *Web Service* que é responsável pela autorização de acesso do utilizador, um mini Site para simular uma plataforma de *eLearning* com um *Web Service* com a funcionalidade de responder se determinado estudante já fez o trabalho de casa e um pequeno Web Site (*MyAmazon*) para apenas ler o conteúdo do cookie para simular o SSO.

Como se pode ver no diagrama da Figura 46, o Subscritor do serviço ADSL, neste caso o pai, cria uma identidade virtual para o filho, a identidade "estudante", e entrega ao filho, que já tem uma identidade virtual "filho" previamente criada pelo pai. Esta identidade virtual "estudante" permite identificar o filho como estudante da sua escola e que lhe permite apenas a aceder ao site de *eLearning* da sua escola. Se ele em vez de usar a identidade "estudante" usar a identidade "filho", já poderá aceder a outros sites, mas só após uma consulta da *Home Gateway* ao site de *eLearning* para verificar se já efectuou todos os trabalhos de casa. Se estiver tudo bem, o acesso com a identidade "filho" é permitido. Os dados referentes à identidade virtual são guardados num cookie que permitirá que o filho seja reconhecido noutro site do domínio do ISP, por exemplo o *MyAmazon*, um site de vendas do ISP.

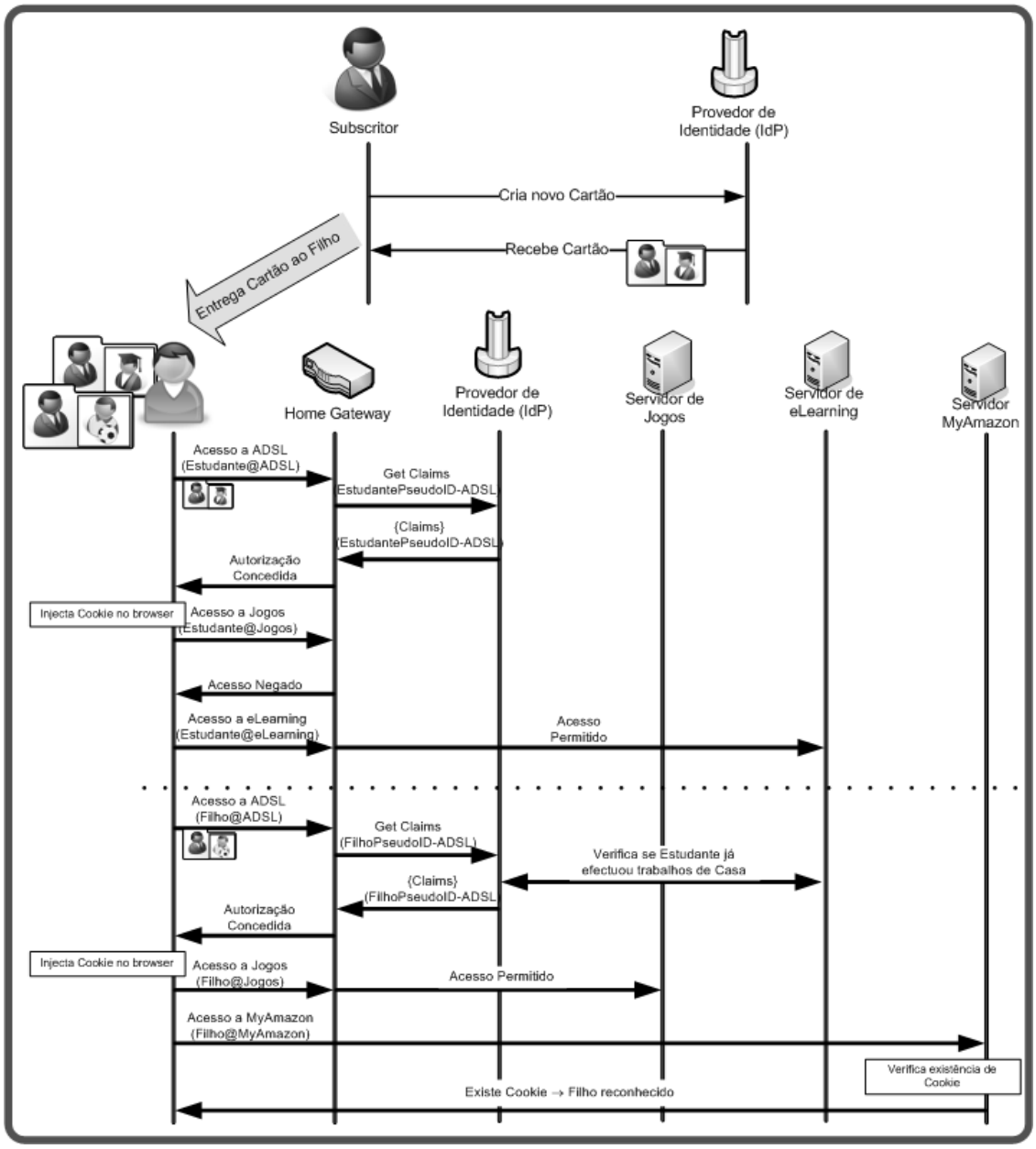


Figura 46 – Demonstração do cenário 2

Com este cenário foi possível demonstrar as capacidades de toda a arquitectura e do protótipo MyIdPv1.5 aplicado num possível cenário de um operador de telecomunicações.

Neste Capítulo apresentou-se o protótipo My Identity Provider desenvolvido nesta Dissertação, evidenciado as suas inovações comparativamente com as soluções que existem actualmente no mercado, que possibilitam o uso de um Provedor de Identidades em cenários como o que foi implementado.

No Capítulo seguinte faz-se uma breve descrição de três arquitecturas de rede – IMS, 3GPP e TISPAN – apresentado uma solução para a integração da gestão de identidades em cada uma destas redes.

6 Extensões de mecanismos e soluções AAA para suporte de Gestão de Identidades em redes de próxima geração

Um operador de rede e, em geral, qualquer entidade que aloje de uma forma segura a informação do utilizador e efectue a sua gestão, pode trazer mais valor ao seu negócio, tendo as vantagens que a Gestão de Identidades pode trazer. Fornecendo uma interoperabilidade entre a sua rede e sistemas de Gestão de Identidades externos, um operador pode aumentar as capacidades da sua rede e características associadas, aproveitando todas as possibilidades que comunicações seguras, gestão e utilização da identidade do utilizador lhe pode trazer.

Neste Capítulo é apresentada uma breve descrição de três arquitecturas de rede – IMS, 3GPP e TISPAN – para de seguida apresentar uma solução para a integração da gestão de identidades em cada uma destas redes.

6.1 IMS

O IP *Multimedia Subsystem* (IMS) [37] é uma arquitectura de uma infra-estrutura que permite aos operadores de rede fornecer aos seus clientes serviços multimédia, baseados em ou desenvolvidos sobre aplicações Internet. Inicialmente criado para redes móveis pelo consórcio 3GPP, evoluiu para poder englobar as redes de próxima geração *Next Generation Networks* (NGN)[38] e a convergência entre rede fixa e móvel. O IMS é agora uma norma comum para as redes de próxima geração, suportando a prestação de serviços multimédia sobre qualquer tecnologia de acesso (*wireless*, *wired* ou móvel), permitindo ao mesmo tempo às operadoras e provedores de serviços, o controlo suficiente para criar, gerir e taxar por esses mesmos serviços. O IMS, quando integrado numa arquitectura de rede de próxima geração, permite a um utilizador receber numa única sessão, e para qualquer dispositivo, conteúdos a partir de múltiplas fontes de aplicação, preservando simultaneamente a sua mobilidade em todos os momentos. Um dos aspectos centrais do IMS é a convergência dos protocolos Internet para a entrega de serviços multimédia em todas as redes de acesso, nomeadamente o IP para transporte de dados e o *Session Initiation Protocol* (SIP) [38] para a negociação e gestão de sessões.

A arquitectura IMS é composta por diversos elementos e respectivas interligações - Figura 47, que operam sobre três camadas funcionais diferentes:

- Camada de Acesso e Transporte (*Transport and Endpoint Layer*)
- Camada de Controlo e Sessão (*Session and Control Layer*)
- Camada de Aplicações/Serviços (*Application/Service Layer*)

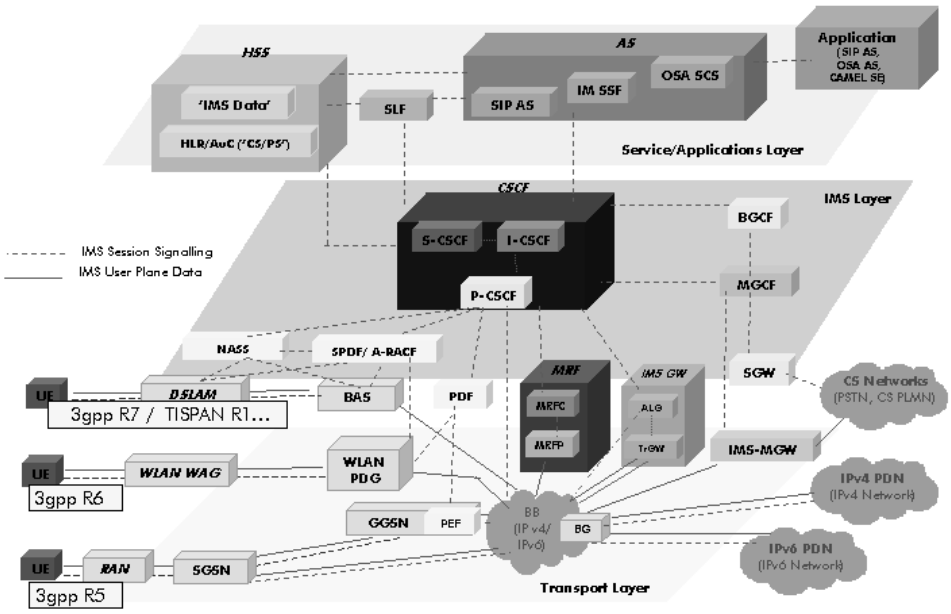


Figura 47 - Visão geral da arquitectura IMS [74]

A ***Camada de Acesso e Transporte*** contém todos os elementos de rede necessários para o transporte da informação com carácter fim a fim, entre utilizadores e plataformas de serviço.

A ***Camada de Controlo e Sessão*** é o núcleo da rede IMS, onde a autenticação, o SIP *routing*, e o controlo de sessão é realizado. Esta camada contém o *Call Session Control Function* (CSCF), que é composto por três entidades diferentes: o servidor *Call Session Control Function* (S-CSCF), o *Proxy Call Session Control Function* (P-CSCF), e o *Interrogating Session Control Function* (I-CSCF). Estes são essencialmente os SIP *proxy servers* encarregues da sinalização SIP IMS (IMS SIP *signalling*). Estes são responsáveis por acções como o SIP *routing*, a autenticação do utilizador, políticas de controlo, autorização de recursos (QoS), geração de registos de tarifação, etc.

A ***Camada de Serviços e Aplicações*** é a camada onde aplicações e servidores de conteúdos (*Applications Servers* – AS's) estão alojados e onde são executados serviços de valor acrescentado para o utilizador. É também nesta camada que todos os dados relacionados com os subscritores estão relacionados, incluindo os identificadores dos utilizadores, perfis de serviço, informação de autenticação e autorização, localização, etc. O elemento responsável por guardar toda esta informação é *Home Subscriber Server* (HSS).

6.1.1 Integração da Gestão de Identidades com a arquitectura IMS

Para conseguir obter as funcionalidades (identidades virtuais, *Single Sign-On*) de um sistema de Gestão de Identidades na arquitectura IMS, serão necessárias modificações na camada de serviços e aplicações. Estas modificações não devem alterar os mecanismos normalizados do IMS, mas estende-los de uma forma coerente.

A melhor forma de efectuar estas modificações com o mínimo de impacto, é acrescentar novas funcionalidades nos elementos e interfaces do IMS. Será uma solução distribuída, isto é, em vez de termos um provedor de identidades, onde são geridas as identidades virtuais e seus atributos, para permitir autenticação ou outro tipo de serviços, aqui tem que se distribuir por alguns elementos funcionais do IMS.

A figura seguinte mostra a correspondência entre as entidades IMS existentes e os conceitos de gestão de identidades.

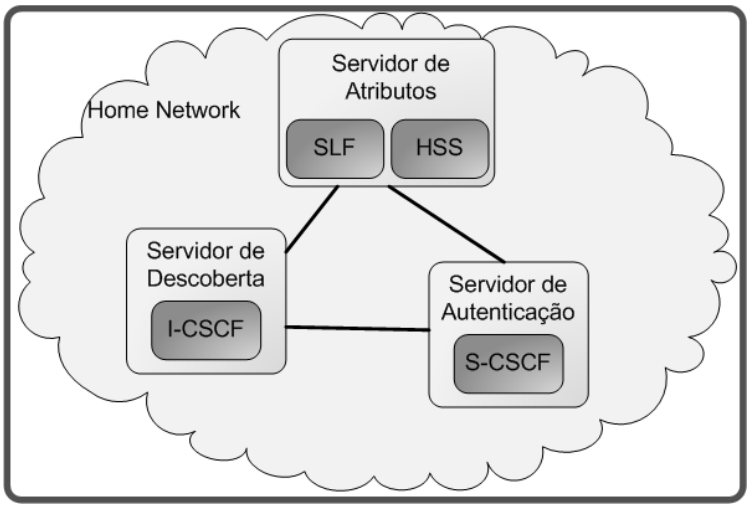


Figura 48 – Correspondência entre as entidades IMS existentes e os conceitos de gestão de identidades

A seguir apresentam-se algumas das características das entidades funcionais apresentadas na Figura 48.

Serving Call Session Control Function (S-CSCF)

- É um SIP *server*, que faz também o controlo de sessão. Está sempre localizado na *home network*. Usa interfaces Diameter Cx e Dx para efectuar o download e upload de perfis do utilizador – não tem armazenamento local para o utilizador. Todas as informações necessárias são descarregadas a partir do HSS.
- Controla os registos SIP, que permite a correspondência da localização do utilizador (ex. endereço IP do terminal) com o endereço SIP.
- Todas as mensagens de sinalização passam por este servidor, podendo inspeccionar qualquer mensagem.
- Decide para qual AS a mensagem SIP será transmitida, com o fim de prestar o seu serviço.
- Fornece serviços de *routing*.

- Aplica as políticas do operador de rede.
- Pode haver múltiplos S-CSCFs numa rede para uma distribuição da carga e também para uma maior disponibilidade. É o HSS que atribui um S-CSCF a um utilizador, quando é consultado pelo I-CSCF.

Interrogating Session Control Function (I-CSCF)

- Localizado no topo do domínio administrativo
- Reencaminha mensagens SIP para outros domínios
- Primeiro contacto na *home network*
- Atribui um S-CSCF ao subscritor que se está a efectuar o registo SIP, consultando o HSS para as decisões de parametrização necessárias

Home Subscriber Server (HSS)

- Base de dados principal, incluindo, entre outros itens, serviço de perfis, informações de localização, informações de segurança (autenticação e autorização) e as identidades do subscritor
- Gestão de mobilidade através do Circuit switching, Packet switching e domínios IMS
- Suporte para o provisionamento de serviços
- Suporte para o estabelecimento de chamada/sessão
- Autorização de serviços

Subscriber Location Function (SLF)

- Localiza o HSS do subscritor quando múltiplos HSS's estão presentes na *home network*

Na arquitectura IMS os dados do utilizador são geridos pelo operador e guardados no HSS. O utilizador não pode facilmente, ou não pode mesmo, alterar essa informação ou atributos. O utilizador não pode definir novas identidades virtuais ou alterar as políticas associadas a essas identidades. O HSS tem o papel de um **Servidor de Atributos**. As capacidades actuais do S-CSCF são as de **Servidor de Autenticação e Autorização**. O I-CSCF tem o papel de descobrir qual a localização do S-CSCF apropriado, sendo o **Servidor de Descoberta**. Para que o utilizador possa gerir as suas Identidades Virtuais tem que ser, então, adicionado um novo elemento. Este elemento será um **Agregador de Identidades**. Este elemento não guarda informação, mas é o elemento que coordena toda a informação do utilizador, isto é, ele não a guarda mas sabe onde pode estar guardada e podendo estar esta num servidor de atributos exterior à rede.

Na Figura 49 pode ver-se a adição do Agregador de Identidades.

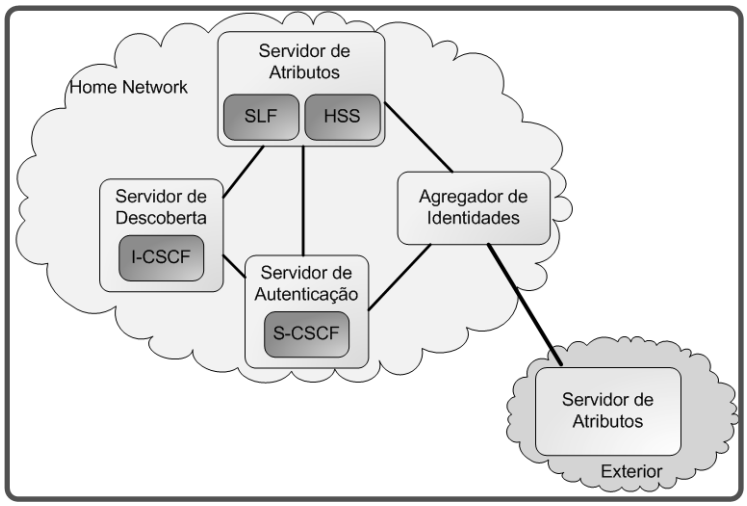


Figura 49 - IMS com suporte para Gestão de Identidades

As interfaces entre o SLF, HSS, S-CSCF e I-CSCF, teriam que ser alteradas de forma a permitir as funcionalidades acima desejadas. Para além disso, para suportar identidades virtuais e políticas de segurança de gestão de identidades o HSS e o modelo de dados de Identidade do IMS teria que ser modificado e as suas interfaces externas estendidas. Para suportar login único, a interface entre o utilizador e S-CSCF também teria que ser alterada, para permitir o envio de *assertions* do agregador de identidades para o utilizador.

6.2 3GPP

Uma rede 3GPP [42] é dividida em duas partes lógicas, que são conhecidas em termos genéricos como a rede de núcleo (*Core Network* – CN) e rede de acesso (*Access Network* – AN). A CN consiste na sobreposição de dois domínios: domínio *Circuit-Switched* (CS) e domínio *Packet-Switched* (PS). Geralmente, as chamadas de voz são sempre geridas pelos componentes pertencentes ao domínio CS. As entidades no domínio PS transportam os dados do utilizador na forma de pacotes autónomos, que são encaminhados independentemente uns dos outros. Assim consegue-se ultrapassar as limitações das redes 2G para a transmissão eficiente de dados. É através da CN que o utilizador pode configurar uma ligação para e de uma rede de pacotes de dados externa (ex: Internet), redes PSTN (*Public Switch Telephone Networks*) e outras redes *wireless*. O domínio AN consiste em dois grandes subsistemas: o UMTS *Terrestrial Radio Access Network* (UTRAN) que foi concebido para suportar ligações de redes UMTS/3G, e o GSM *EDGE Radio Access Network* (GERAN) que fornece suporte para ligações móveis GSM. A figura seguinte mostra a arquitectura da rede 3GPP.

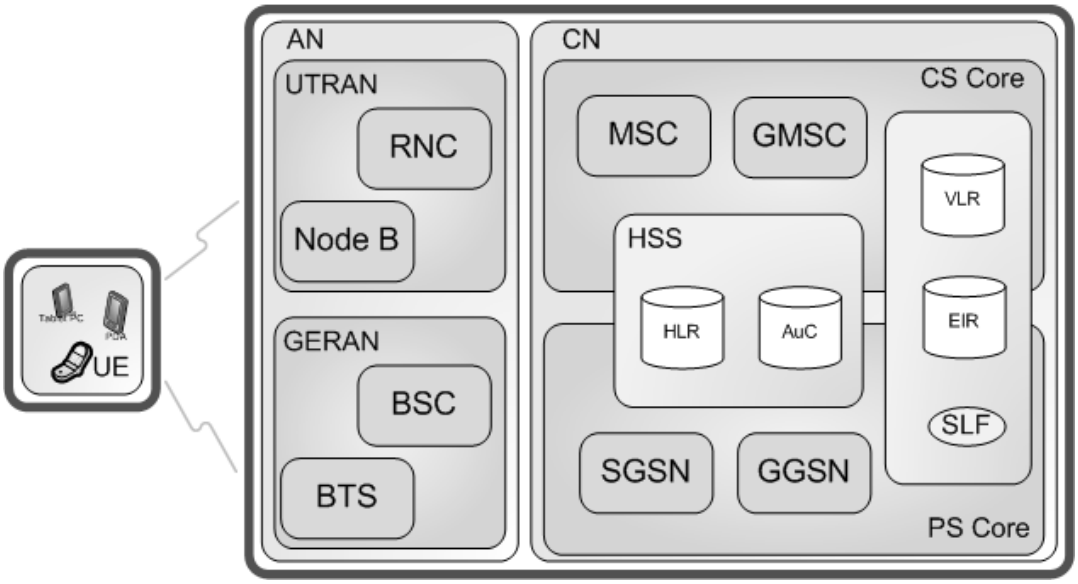


Figura 50 - Arquitectura 3GPP

6.2.1 Integração da Gestão de Identidades com redes 3GPP

As redes 3GPP têm funcionalidades reduzida em Gestão de Identidades, as transacções de informação estão limitadas ao domínio local, ou seja, os atributos dos utilizadores são obtidos apenas dos repositórios locais e não podendo estes atributos serem também fornecidos a domínios exteriores.

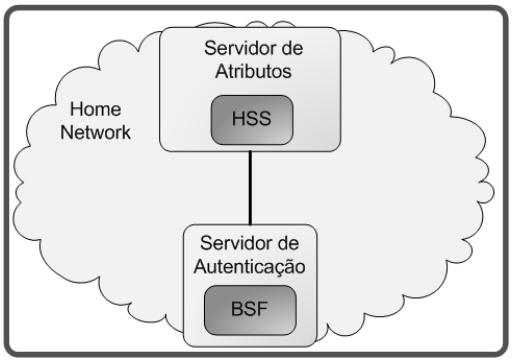


Figura 51 - Correspondência entre as entidades 3GPP existentes e os conceitos de gestão de identidades

A seguir apresentam-se algumas das características das entidades funcionais apresentadas na Figura 51.

- ***Home Subscriber Server (HSS)***: O HSS é a base de dados principal para um dado utilizador. É a entidade que contém toda a informação relacionada com a subscrição para suportar as entidades da rede que suportam actualmente o manuseamento chamadas/sessões. O HSS armazena as definições de segurança do GBA [45] do utilizador (GBA *user security settings* – GUSS's).
- ***Bootstrapping Server Function (BSF)***: O BSF é o elemento que fornece funções independentes da aplicação, para permitir a mútua autenticação entre o equipamento do utilizador e um servidor que antes não se conheciam. [45]

De forma a possibilitar a interacção com outros domínios, a gestão de identidades pode dar uma ajuda. O utilizador deve ter a possibilidade de poder usar diferentes identidades, cada uma com os seus atributos (identidades virtuais). Deve ser possível usar informação de outras fontes exteriores ao domínio nas suas identidades virtuais. Deve também controlar como é que toda esta informação de identidade é distribuída pelos diferentes serviços e entidades. Login único.

Para permitir que um utilizador possa ter e definir múltiplas identidades virtuais, as quais poderão ter diferentes atributos, é necessário um adicionar um elemento que agregue estas identidades, como no caso anterior do IMS – Agregador de Identidades. Este Agregador de Identidades deve estar colocado junto do BSF, pois as funcionalidades do BSF são todas aproveitadas por este agregador de identidades. O agregador de identidades tem também a capacidade de ir buscar atributos não só ao HSS mas também fora do domínio. Por sua vez as funcionalidades do HSS teriam que ser estendidas de forma a permitir que outros serviços externos pudessem usar este Servidor de Atributos.

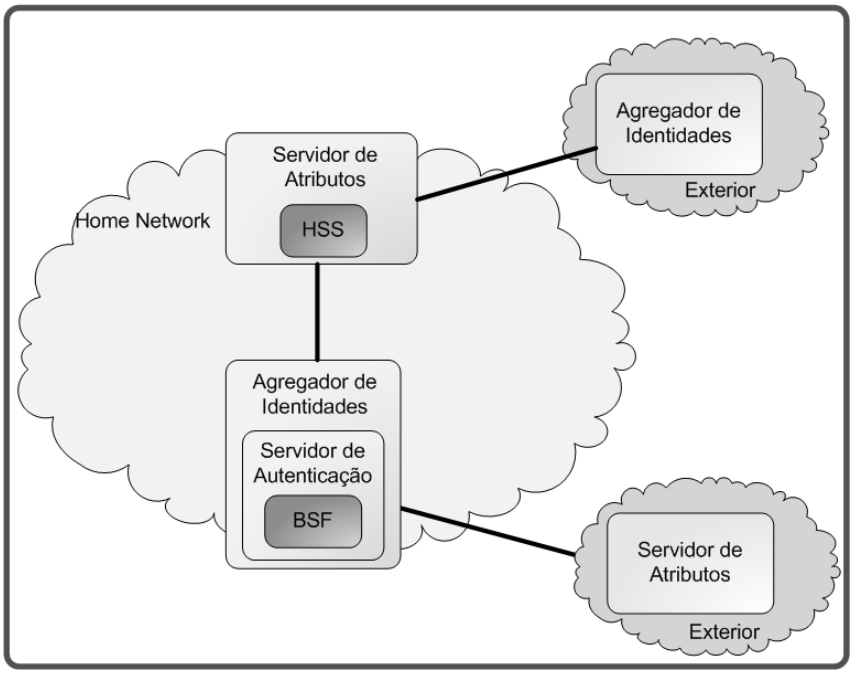


Figura 52 - 3GPP com suporte para Gestão de Identidades

Algumas interfaces teriam que ser modificadas, estendidas, nomeadamente a interface entre o BSF e o HSS, e a interface entre o BSF e o equipamento do utilizador, de forma a permitir as funcionalidades pretendidas.

6.3 ETSI TISPAN Transport layer Architecture

Em 2003 ETSI e o 3GPP iniciaram esforços conjuntos na pesquisa de como harmonizar o IMS *core* para redes *wireless* e *wireline*. Deste trabalho resultou o TISPAN *Next Generation Network* (NGN) *Release* 1 [50], finalizado em 2005. No início de 2008, a *Common* IMS *Specification* foi transferida de volta para o 3GPP, sendo a única organização de normalização responsável por fornecer uma *Common* IMS ajustada a qualquer rede. O TISPAN continuou a trabalhar na especificação NGN focando-se em adicionar elementos chave ao NGN como IPTV baseado em IMS e não IMS, *Home Networks*, dispositivos, assim como interligações NGN entre Redes Corporativas. Este esforço resultou no TISPAN NGN *Release* 2 [51], publicado no início 2008. Actualmente, o TISPAN começou a desenvolver a 3ª *release* das especificações NGN com especial atenção em melhoramentos no IPTV, interligações entre redes IP, melhoramentos na segurança NGN e QoS com controlo de sobrecarga, entre outras exigências necessitando de constante evolução.

As especificações do TISPAN's NGN têm como principal meta uma plataforma extensível, para o desenvolvimento de futuros serviços e arquitecturas. Tem como dois objectivos principais: estender os serviços no IMS para o múltiplo acesso a redes baseadas em diferentes tecnologias e capacidades (ex. xDSL, *Ethernet*, redes cabo ou *wireless*) e substituir/emular redes ISDN/PSTN. A Figura 53 apresenta uma arquitectura geral do TISPAN NGN.

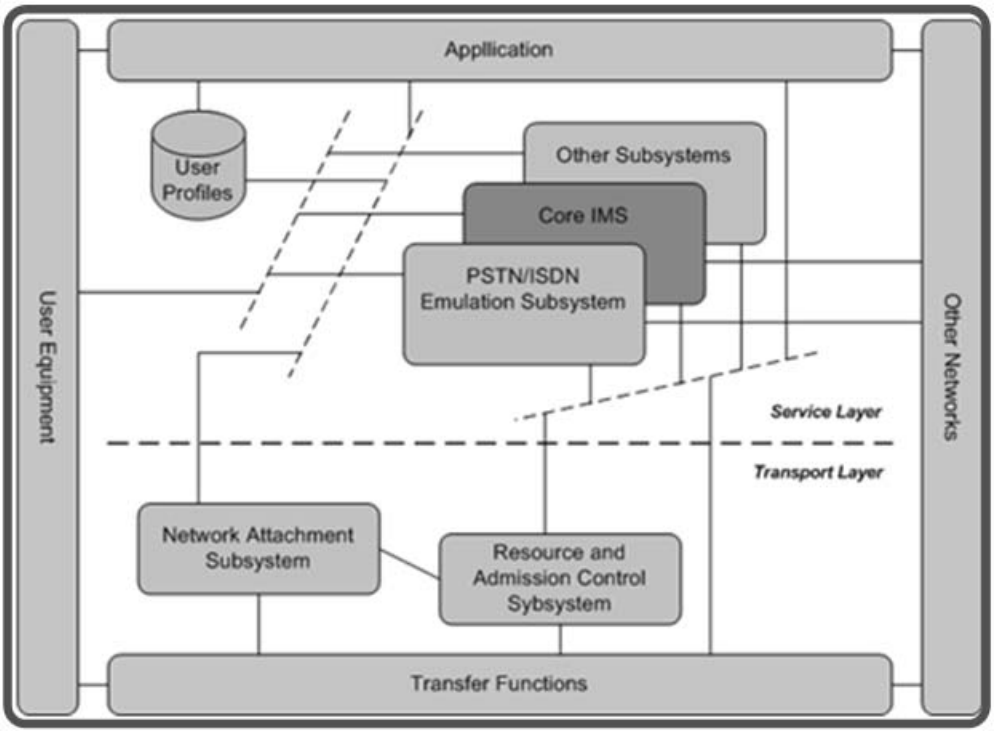


Figura 53 - Arquitectura geral do TISPAN NGN

A arquitectura TISPAN é um conjunto de subsistemas e entidades funcionais localizadas em duas camadas funcionais diferentes: a camada de Transporte e a camada de Serviço.

A ***camada de Transporte*** fornece conectividade IP ao equipamento do utilizador. As funcionalidades desta camada estão divididas em duas sub-camadas, uma de transporte e outra de controlo. A sub-camada de transporte executa funções como reencaminhamento e *routing* de pacotes. Por outro lado a sub-camada de controlo, que esconde a complexidade das várias tecnologias de transporte usadas no acesso e das redes de core (abaixo da camada IP), é composta por dois elementos principais; o RACS (*Resource Admission Control Subsystem*) [52] e o NASS (*Network Attachement Subsystem*) [53].

A ***camada de Serviço*** consiste num conjunto de subsistemas que oferece funcionalidades de controlo de serviço. Entre eles está o Core IMS [54], que fornece meios para negociar serviços multimédia baseados em SIP a terminais NGN. Os restantes elementos da camada de Serviço são:

- Subsistema de Emulação PSTN/ISDN;
- Outros subsistemas multimédia (ex. Subsistema dedicado IPTV) e aplicações;
- Componentes comuns, usados por diversos subsistemas, como os componentes necessários para aceder a aplicações, funções de taxação, gestão de perfis de utilizador, gestão de segurança, bases de dados de *routing*, etc.

Esta arquitectura permite a adição de novos subsistemas ao longo do tempo, cumprindo novas exigências e classes de serviço. A conectividade IP é fornecida ao equipamento NGN do utilizador pela camada de transporte, debaixo do controlo do NASS e do RACS.

O TISPAN NGN apresenta na sua génese, um conjunto de requisitos e características. Em relação à implementação da camada de transporte aplica-se o seguinte:

- O operador NGN deve ter uma identidade única para cada utilizador, terminal, rede, elemento, etc, independentemente do serviço ou tecnologia usada;
- Uma identidade é um conjunto de dados, pertencentes a um utilizador, que deve ser mantido pelo operador numa forma escalável e altamente segura, mesmo após a remoção da conta.
- Utilizadores devem experienciar mobilidade pessoal e/ou de terminal, *roaming* e nomadismo.
- Privacidade, confidencialidade e integridade deve ser suportada.
- Políticas devem ser aplicadas nos elementos de transporte baseado em perfis de utilizador.

6.3.1 Integração da Gestão de Identidades com redes TISPAN

Para adicionar funcionalidades de Gestão de Identidades, são necessárias algumas modificações na arquitectura da camada de transporte do TISPAN. Estas modificações não devem alterar o comportamento normalizado dos mecanismos do TISPAN, mas estende-los de uma forma consistente.

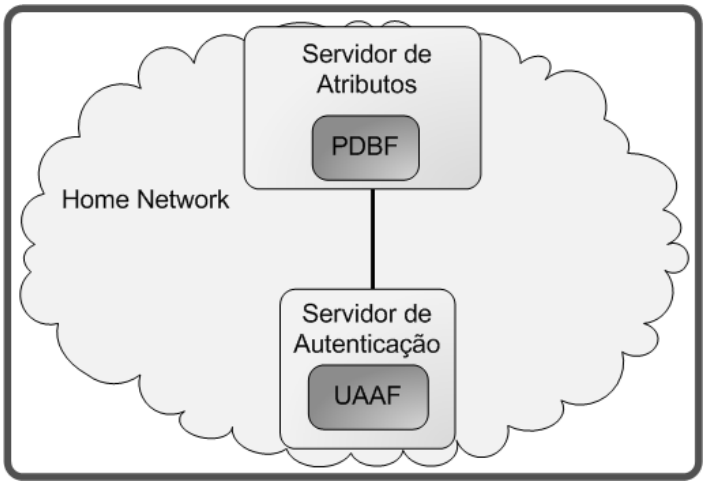


Figura 54 – Correspondência entre as entidades TISPAN existentes e os conceitos de gestão de identidades

A seguir apresentam-se algumas das características das entidades funcionais apresentadas na Figura 54.

- ***Profile Database Function (PDBF):*** é a entidade funcional que contém os dados de autenticação do utilizador (ex. identidade do utilizador, lista de métodos de autenticação suportados, chaves de autenticação, etc) e informação relacionada com a configuração do acesso à rede pretendida – os perfis de rede do utilizador.
- ***User Access Authorization Function (UAAF):*** efectua a autenticação do utilizador, assim como a verificação da autorização, baseada em perfis de utilizador, para acesso à rede. Para cada utilizador, o UAAF recebe os dados de autenticação e a informação da autorização de acesso do perfil de rede do utilizador contido no PDBF. O UAAF, também receber dados de contabilização para a taxação do serviço usado. O UAAF actuando como um *proxy* pode localizar e comunicar com o UAAF que actua como *server*, o qual pode consultar os dados de autenticação do utilizador armazenados no PDBF, e reencaminhar pedidos de acesso e autorização, assim como mensagens de contabilização.

Na arquitectura actual do TISPAN, os dados do utilizador são geridos pelo operador. É impossível ao utilizador alterar a sua informação em tempo real. Sendo assim, o utilizador não pode definir identidades virtuais.

As identidades virtuais, como já visto, podem ser compostas por diferentes informações do utilizador (atributos), permitindo que o utilizador se apresente com diferentes identidades durante a autenticação na rede. O PDBF deve estar colocado com o Servidor de Atributos, para que seja possível estender o modelo de dados do PDBF para suportar as funcionalidades de Gestão de Identidades. Assim, será possível definir políticas de privacidade, controlar a informação do utilizador que é disponibilizada a terceiros.

A Figura 55 apresenta uma nova adição ao cenário de forma a suportar o uso de atributos do utilizador de Provedores de Identidade externos à rede, havendo assim a possibilidade do utilizador ter a sua identidade espalhada por diferentes operadores.

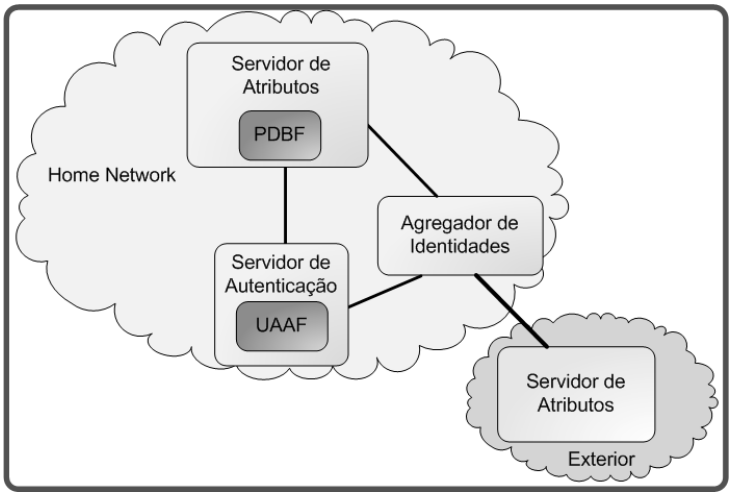


Figura 55 - TISPAN com suporte para Gestão de Identidades

O elemento Agregador de Identidades é introduzido para permitir ir buscar identidades e/ou atributos do utilizador a outras redes.

Têm que ser definidas novas interfaces na camada de transporte do TISPAN, enquanto outras existentes terão que sofrer ligeiras modificações. A interface entre o PDBF e o UAAF tem que ser alterada para adicionar as novas funcionalidades. Para adicionar as novas funcionalidades é necessário alterar também o modelo do PDBF, assim como as interfaces para o exterior.

Neste Capítulo fez-se uma pequena descrição de cada uma das arquitecturas de rede, tendo sido apresentada uma solução para a integração da gestão de identidades em cada uma delas.

7 Conclusões e considerações finais

7.1 Reflexão Crítica

A Identidade é formada por um conjunto de atributos que descrevem uma entidade, estando muitos atributos armazenados em subconjuntos – identidades virtuais - que estão espalhados por vários repositórios – servidores de atributos – de redes diferentes. Novos formatos de dados são criados em aplicações e sistemas, baseadas em normas proprietárias, incompatíveis com sistemas e aplicações de outros fabricantes, sendo armazenados em sistemas completamente fechados a qualquer tipo de comunicação com outros sistemas ou comunicando de uma forma muito limitada.

Para um fabricante este tipo de sistemas proprietário até pode ser uma boa ideia, porque obriga os seus clientes a terem de continuar a adquirir os seus produtos, se quer que o seu sistema cresça. Mudar de sistema para o de outro fabricante acarreta sempre grandes despesas, o que leva a que muitas vezes, muitas organizações, não estando satisfeitas com a qualidade ou suporte do produto que compraram, são obrigados a manter-se com determinado fabricante devido aos custos elevados dessa mudança.

Existem inúmeras soluções proprietárias para efectuar as tarefas relacionadas com a Gestão de Identidades. A Gestão de Identidades é o caminho para o provisionamento, manutenção, segurança, privacidade, troca e controlo das identidades. Um dos principais propósitos da Gestão de Identidades é ter sempre informação actualizada para que a Autorização de Acesso a determinados recursos seja sempre feito de uma forma controlada e segura.

A existência de uma plataforma comum para a troca de informação de identidade cria novos serviços, novos oportunidades de negócio. Um factor de sucesso para um negócio, é a habilidade para estabelecer relações de longa duração com os seus clientes. A melhor forma de manter essa relação é fornecer produtos ou serviços o mais personalizados possíveis. Sem uma plataforma de Gestão de Identidades baseada numa infra-estrutura comum para uma troca segura de informação de identidade, não será possível descobrir e manter dados com informação das preferências dos seus clientes.

Nesta Dissertação identificaram-se quais as normas e protocolos existentes e idealizaram-se alguns cenários possíveis de serviços usando a Gestão de Identidades. Estes poderão ser implementados por uma operadora de telecomunicações, tendo como principal objectivo fornecer serviços mais personalizados ao cliente, trazendo também mais valor à operadora. Muitos outros cenários poderão ser idealizados, em situações tão diversas, quantas se possam imaginar. Actualmente a nível da internet as tecnologias mais usadas são o OpenID e o *Identity Metasystem*, sendo por isso de prever que de todas as tecnologias estudadas venha a ser essa com mais utilização no futuro, daí se ter optado pelo *Identity Metasystem* aquando da

implementação dos cenários. A Gestão de Identidades ainda está a dar os primeiros passos na sua aplicação no “dia-a-dia”.

As soluções de Gestão de Identidades existentes no mercado estão mais direccionadas para a gestão de atributos pessoais (nome, morada, idade,etc), para possibilitar o registo e autenticação em aplicações e serviços, não prevendo outro tipo de utilização sendo assim bastante limitadas.

Utilizando este tipo de soluções para além da gestão de atributos do utilizador, pode-se como foi demonstrado com o protótipo desenvolvido (MyIdPv1.5) aplicar este tipo de tecnologias noutro tipo de cenários de forma a dar ao utilizador novos serviços, privacidade e segurança, permitindo às diversas entidades e organizações novas formas de negócio.

Este tipo de soluções ainda é muito pouco divulgado, sendo utilizado de uma forma muito localizada, não havendo um conhecimento pelo utilizador comum deste tipo de tecnologias. Quem terá uma maior responsabilidade em dar a conhecer e fornecer serviços baseados neste tipo de tecnologia, serão as operadoras de telecomunicações.

O desenvolvimento do protótipo exigiu uma curva de aprendizagem das linguagens e tecnologias envolvidas acentuada e rápida para poder estar disponível em tempo útil. Neste momento está a ser estudada e protótipada uma demonstração para a sua integração com um sistema de IPTV de uma grande operadora nacional.

Nesta Dissertação fez-se também um estudo dos elementos responsáveis por armazenar, autenticar e aplicar políticas dos seus utilizadores em cada uma das redes de próxima geração, para que se pudesse idealizar uma forma de integrar as funcionalidades de Gestão de Identidades.

7.2 Trabalho futuro

Actualmente existe ainda muita discussão acerca de algumas definições sobre Gestão de Identidades e que papeis devem ter as diversas entidades intervenientes em todo o processo (provedor de identidades, provedor de serviços, etc). Por exemplo, actualmente existe alguma discussão se um provedor de identidades deve ou não guardar políticas, para além de atributos do utilizador, ou seja, em vez de as políticas estarem definidas em cada um dos provedores de serviço, se não poderiam estar no provedor de identidades e o provedor de serviço apenas as aplicava. Isto vai contra os conceitos definidos, mas na óptica de muitos poderá trazer vantagens, visto o utilizador não ter que definir as políticas de privacidade e acesso em cada um dos provedores de serviço, o que pode tornar o processo moroso e algo aborrecido. Esta questão é importante, porque a maneira como se poderão idealizar cenários no futuro seguindo as duas hipóteses é bastante diferente. Era importante para manter um sistema de Gestão de Identidades o mais normalizado possível para que pudesse ter uma utilização quase ou até mesmo universal que se chegasse a um consenso.

Como trabalho futuro, seria interessante implementar o MyIdPv2.0 que para além das características dos seus antecessores teria a capacidade de se adaptar automaticamente a novos serviços que fossem criados. Esta seria uma primeira aproximação à criação de um Provedor de Identidades com possibilidade de utilização por qualquer serviço. Seria também bastante útil implementar o suporte para outras tecnologias para além do *Identity Metasystem* (exemplo *OpenID*, *Shibboleth*), mas mantendo as características que permitem a sua aplicação por exemplo aos cenários apresentados nesta Dissertação. Assim teríamos um Provedor de Identidades com enormíssimas potencialidades no mercado, o que actualmente não acontece.

Neste momento está a fazer-se um estudo para a aplicação futura do Cenário 4 do Capítulo 4 "Acesso a serviço através de autenticação/autorização *Out-of-band*", que poderá também ainda dar origem a outro tipo de cenários passíveis de serem implementados com alguma simplicidade por operadoras de telecomunicações.

As redes de próxima geração são uma certeza. A Gestão de Identidades traz mais-valias aos operadores que a integrarem nas suas redes e aos seus clientes. Será portanto necessário adaptar as redes de próxima geração a normas abertas, para que todos possam competir ao mesmo nível. Seria também importante fazer um estudo aprofundado das alterações necessárias a fazer nos diversos elementos e interfaces de cada uma das arquitecturas para se poder passar para a criação de um protótipo de uma implementação de um sistema de Gestão de Identidades em cada uma das redes de próxima geração.

A Gestão de Identidades é uma obrigação para o presente e para o futuro.

8 Referências

[1] http://jkobielus.blogspot.com/2005/11/imho-identity-privacy-reputation.html

[2] Ping Identity Corporation , http://www.pingidentity.com/

[3] Windley, Phillip J. Digital Identity. O'Reilly Media, 2005

[4] Simple Object Access Protocol (SOAP), http://www.w3.org/TR/soap/

[5] SOAP Version 1.2 specification, http://www.w3.org/TR/soap12-part1/

[6] OASIS Security Services (SAML) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[7] SAML Specifications, http://saml.xml.org/saml-specifications

[8] OASIS eXtensible Access Control Markup Language (XACML) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[9] eXtensible Access Control Markup Language (XACML) Version 2.0 Specification http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[10] OASIS Provisioning Services TC, http://www.oasis-open.org/committees/provision/

[11] OASIS Provisioning Services Specifications, http://www.oasis-open.org/committees/download.php/17708/pstc-spml-2.0-os.zip

[12] OASIS Web Services Security (WSS) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[13] Web Services Security: SOAP Message Security 1.1 - http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

[14] OASIS Web Services Secure Exchange (WS-SX) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-sx

[15] WS-Trust 1.4 Specification, http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.pdf

[16] Web Services Metadata Exchange Specification - http://www.w3.org/TR/ws-metadata-exchange/

[17] OASIS, http://www.oasis-open.org

[18] Cover Pages, http://xml.coverpages.org/

[19] XML.org, http://www.xml.org/

[20] Liberty Alliance, http://www.projectliberty.org

[21] Liberty Alliance Specifications, http://www.projectliberty.org/specifications__1

[22] OpenID, http://openid.net/

[23] OpenID Authentication 2.0 – Final - http://openid.net/specs/openid-authentication-2_0.html

[24] Identity Blog, http://www.identityblog.com/

[25] OASIS Identity Metasystem Interoperability (IMI) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=imi

[26] Visão da Microsoft sobre o Identity Metasystem, http://msdn.microsoft.com/en-us/library/ms996422.aspx

[27] Leis da Identidade, http://msdn2.microsoft.com/en-us/library/ms996456.aspx

[28] Identity Metasystem Interoperability Version 1.0, http://docs.oasis-open.org/imi/identity/v1.0/cs/identity-1.0-spec-cs-01.html

[29] Informations Cards, http://informationcard.net/quick-overview
[30] Information Card Ecosystem white paper, http://72.51.47.212/files/icf-information-card-ecosystem-white-paper.pdf
[31] Understanding Personal Information Cards, http://msdn.microsoft.com/en-us/library/aa347717.aspx
[32] Managing Information Cards with Windows CardSpace, http://msdn.microsoft.com/en-us/library/aa347713.aspx
[33] Identity Selector Interoperability Profile V1.0 http://download.microsoft.com/download/1/1/a/11ac6505-e4c0-4e05-987c-6f1d31855cd2/Identity-Selector-Interop-Profile-v1.pdf
[34] Open ID Information Card, https://openidcards.sxip.com/
[35] OpenID Information CardSspecifications, https://openidcards.sxip.com/spec/openid-infocards.html
[36] Identity Management for Converged Net-works", Lucent/Sun Whitepaper, 2006
[37] 3GPP TS 23.228 V8.7.0: "IP Multimedia Subsystem (IMS); Stage 2", 2008
[38] IETF RFC 3261: "SIP: Session Initiation Protocol", 2002
[39] IETF RFC 2486: "The Network Access Identifier", 1999
[40] IETF RFC 2396: "Uniform Resource Identifiers (URI): Generic Syntax", 1998
[41] IETF RFC 3966: "The tel URI for Telephone Numbers", 2004
[42] 3GPP TS 23.002 V8.4.0: "Network Architecture", 2008
[43] 3GPP TS 33.102 V8.1.0: "3GPP Security; Security architecture", 2008
[44] 3GPP TR 33.919 V8.0.0: "Generic Authentication Architecture (GAA); System Description", 2008
[45] 3GPP TS 33.220 V8.5.0: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture", 2008
[46] 3GPP TS 33.221 V8.0.0: "Generic Authentication Architecture (GAA); Support for subscriber certificates", 2008
[47] http://www.cs.huji.ac.il/~sans/students_lectures/GSM%20Attacks.ppt
[48] ITU-T Recommendation E.164: "The international public telecommunication numbering plan", 2005
[49] 3GPP TS 23.003 V8.1.0: "Numbering, Addressing and Identification", 2008
[50] ETSI ES 282 001 V1.1.1: "NGN Functional Architecture Release 1", 2005
[51] ETSI ES 282 001 V2.0.0: "NGN Functional Architecture", 2008
[52] ETSI ES 282 003 V2.0.0: "Resource and Admission Control Sub-System (RACS): Functional Architecture", 2008
[53] ETSI ES 282 004 V2.0.0: "NGN Functional architecture – Network Attachment Sub-System (NASS)", 2008
[54] ETSI ES 282 007 V2.1.1: "IP Multimedia Subsystem (IMS); Functional architecture", 2008
[55] Extensible Authentication Protocol (EAP), http://tools.ietf.org/html/rfc3748
[56] ETSI TS 187 003 V1.7.1: "NGN Security; Security Architecture", 2008
[57] HTTP - Hypertext Transfer Protocol, http://www.w3.org/Protocols/
[58] Extensible Markup Language (XML), http://www.w3.org/XML/

[59] XML Signature Syntax and Processing (Second Edition), http://www.w3.org/TR/xmldsig-core/
[60] XML Encryption Syntax and Processing, http://www.w3.org/TR/xmlenc-core/
[61] Web Services Policy 1.5 – Framework, http://www.w3.org/TR/ws-policy/
[62] Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl
[63] SGML Resources, http://www.w3.org/MarkUp/SGML/
[64] Shibboleth Web Site, http://shibboleth.internet2.edu/
[65] MACE, http://middleware.internet2.edu/dir/
[66] Shibboleth 1.x, https://spaces.internet2.edu/display/SHIB/WebHome
[67] Shibboleth 2.0, https://spaces.internet2.edu/display/SHIB2/Home
[68] XMLdap, http://xmldap.org
[69] OAuth, http://oauth.net
[70] OAuth Core 1.0, http://oauth.net/core/1.0
[71] Google Data API's, http://code.google.com/intl/pt/apis/accounts/docs/OAuth.html
[72] Identity Selector Interoperability Profile specification and companion guides, http://www.microsoft.com/downloads/details.aspx?FamilyID=B94817FC-3991-4DD0-8E85-B73E626F6764&displaylang=en
[73] WSO2 Identity Solution, http://wso2.org/projects/identity
[74] http://en.wikipedia.org/wiki/File:Ims_overview.png
[75] http://goliath.ecnext.com/coms2/gi_0199-6152030/Notes-toward-the-definition-of.html
[76] http://www.informit.com/articles/article.aspx?p=21417&seqNum=6
[77] Java.sun.com - The Source for Java Developers, http://java.sun.com/
[78] JavaServer Pages Technology, http://java.sun.com/products/jsp/
[79] SQL, http://www.sql.org/
[80] Servlet's, Java Servlet Technology, http://java.sun.com/products/servlet/
[81] Apache Tomcat, http://tomcat.apache.org/

Anexos

1 Protocolos usados em Gestão de Identidades

1.1 SOAP

O protocolo SOAP, *Simple Object Access Protocol* [4], é um elemento fundamental para os *Web Services*.

O protocolo SOAP é um protocolo simples para troca de informação num ambiente distribuído e descentralizado como é o ambiente da Internet. Permite que objectos ou código de qualquer tipo, em qualquer plataforma e em qualquer linguagem, possam comunicar entre si. O SOAP está implementado em mais de 60 linguagens e em mais de 20 plataformas. Assim, um objecto em qualquer lado, local ou remoto, tem a capacidade de comunicar com outros objectos.

Uma grande vantagem do SOAP é que ele foi adoptado pela grande maioria de fabricantes de hardware e software visto a especificação do SOAP ser aberta e disponibilizar uma base para comunicação entre aplicações – *Web Services*.

É um protocolo baseado em XML [58] (*eXtensible Markup Language*), mas mais flexível, que uniformiza o formato das estruturas das mensagens. As mensagens são o meio fundamental para a troca de mensagens entre os *Web Services* e clientes. O uso do XML para a codificação das mensagens SOAP tráz vantagens:

- O XML pode ser facilmente lido por humanos, sendo assim fácil de entender e eliminar erros.
- Existem variadíssimos XML *parsers* amplamente disponíveis.
- O XML é uma norma aberta.
- Permite uma simplificação da especificação.

1.1.1 Funcionalidades do SOAP

O SOAP fornece algumas funcionalidades como sendo as mais significativas a interoperabilidade entre sistemas usando linguagens e protocolos normalizados e amplamente difundidos, como o XML e HTTP (*HyperText Transfer Protocol*) [57], permite também a comunicação entre sistemas protegidos por *firewalls*, sem que seja necessária abertura de portos adicionais, aumentado a segurança, visto que o SOAP usa por norma o porto 80. Outra funcionalidade é a de descrever por completo cada elemento da mensagem, facilitando o entendimento e a detecção de erros.

1.1.2 Mensagens SOAP

Uma mensagem SOAP é composta por três elementos básicos como se pode ver na figura seguinte:

Figura 56 – Estrutura de uma mensagem SOAP

- Envelope (*Envelope*) – É o elemento principal do XML que representa a mensagem.
- Cabeçalho (*Header*) – É um mecanismo genérico de adição de características à mensagem SOAP de uma forma descentralizada sem acordo anterior entre as partes comunicantes.
- Corpo (*Body*) – Contém a codificação actual de uma chamada de um método e todos os argumentos de entrada, ou uma resposta codificada que contém o resultado de uma chamada de um método.

1.1.3 Envelope SOAP

O *Envelope* SOAP é o elemento obrigatório de uma mensagem SOAP. Funciona como um recipiente para todos os outros elementos da mensagem. O *Envelope* SOAP precisa das informações específicas do protocolo de transporte que está ligado a ele, com o intuito de garantir que o envelope é entregue no local certo. No HTTP, existe um cabeçalho *SOAPAction* que indica qual o endereço de entrega da mensagem.

Exemplo de formatação de um *Envelope* SOAP:

```
<soap:Envelope
xmlns:xsi="Schema-Instance"
xmlns:xsd="Schema"
xmlns:soap="Envelope">
      <soap:Body>
            <!-- Elementos do corpo inseridos aqui!!! -->
      </soap:Body>
</soap:Envelope>
```

1.1.4 Header SOAP

O *Header* de uma mensagem SOAP é opcional. O *Header* define meta data que pode fornecer um contexto à mensagem ou redireccionar o processamento da mensagem.

Exemplo da formatação de um *Header* SOAP

```
<soap:Envelope
xmlns:xsi="Schema-Instance"
xmlns:xsd="Schema"
xmlns:soap="Envelope">
    <soap:Header>
        <Authentication xmlns="Local">
            <User>user</User>
            <Password>password</Password>
        </Authentication>
    </soap:Header>
    <soap:Body>
<!-- Elementos do corpo inseridos aqui!!! -->
    </soap:Body>
</soap:Envelope>
```

Se uma mensagem contém um *Header*, este deve ser o primeiro elemento a aparecer na mensagem após a *tag* de abertura do envelope e todos os elementos filhos do *Header* são definidos como *Headers* separados e chamados de entradas do *Header* (*Header entries*).

1.1.5 Body SOAP

O *Body* de uma mensagem SOAP é obrigatório. O *Body* da mensagem guarda os dados específicos de uma chamada de um método, tais como, os parâmetros de entrada, de saída e os resultados produzidos pelo método. As utilizações do corpo da mensagem incluem chamadas remotas a métodos e notificações de erros.

O *Body* da mensagem aparece a seguir ao *Header*, caso este exista. Se não existir, aparece logo a seguir à *tag* de abertura do envelope.

O conteúdo do *Body* depende de a mensagem ser um pedido ou uma resposta. Se for um pedido, ele contém informações sobre a chamada do método. Se for uma resposta, ele contém os dados do resultado da chamada do método.

Exemplo de um pedido SOAP que envia um valor para ser convertido de decimal para binário para *Web Service* localizado em http://convert.myexample.com.

```
<soap:Envelope
xmlns:xsi="Schema-Instance"
xmlns:xsd="Schema"
xmlns:soap="Envelope">
    <soap:Body>
        <Convert xmlns="http://convert.myexample.com ">
            <Value>10</Value>
            <From>DEC</From>
            <To>BIN</To>
        </Convert>
    </soap:Body>
</soap:Envelope>
```

Exemplo da resposta SOAP ao pedido do exemplo anterior para o endereço http://site.mysite.com do serviço que fez o pedido.

```
<soap:Envelope
xmlns:xsi="Schema-Instance"
xmlns:xsd="Schema"
xmlns:soap="Envelope">
    <soap:Body>
        <ConvertResponse xmlns="http://site.mysite.com">
            <ValueResult>1010</ValueResult>
        </ConvertResponse>
    </soap:Body>
</soap:Envelope>
```

1.1.6 SOAP sobre HTTP

Para entregar mensagens codificadas em SOAP, é necessário usar um protocolo de comunicação fim a fim. No caso de *Web Services*, o protocolo mais usado é o HTTP. O SOAP utiliza assim o modelo de pedido/resposta proposto pelo HTTP, baseado no HTTP *POST* e HTTP *RESPONSE*.

O HTTP *POST* é o comando responsável pelo envio da mensagem SOAP. A Figura 57 representa uma estrutura SOAP contida numa estrutura HTTP *POST*.

Figura 57 – Estrutura do http POST com uma mensagem SOAP

Exemplo anterior enviado via HTTP *POST*:

```
POST /ctemp/ctemp.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: " http://site.mysite.com "
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="Schema-Instance"
xmlns:xsd="Schema"
xmlns:soap="Envelope">
    <soap:Body>
        <Convert xmlns="http://convert.myexample.com ">
            <Value>10</Value>
            <From>DEC</From>
            <To>BIN</To>
        </Convert>
    </soap:Body>
</soap:Envelope>
```

O HTTP *RESPONSE* é o comando responsável por devolver a resposta de um *Web Service*. Exemplo da resposta ao HTTP *Post* anterior:

```
HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="Schema-Instance"
xmlns:xsd="Schema"
xmlns:soap="Envelope">
    <soap:Body>
        <ConvertResponse xmlns="http://site.mysite.com">
            <ValueResult>1010</ValueResult>
        </ConvertResponse>
    </soap:Body>
</soap:Envelope>
```

O SOAP, sendo um elemento principal na estrutura dos *Web Services*, é-o também para a Gestão de Identidades. O SOAP é um protocolo que é amplamente usado pelas diversas tecnologias e protocolos de Gestão de Identidades e, consequentemente, há algum interesse em perceber o seu funcionamento para melhor compreensão destas tecnologias e protocolos.

Uma descrição mais detalhada pode ser encontrada em [5]

1.2 SAML

SAML (*Security Assertion Markup Language*) [6], desenvolvido pelo Comité Técnico para Serviços de Segurança da OASIS (*Organization for the Advancement of Structured Information Standards*) [17], é uma estrutura baseada em XML para comunicar informação sobre a autenticação, direitos e atributos de um utilizador. O SAML permite a uma entidade fazer afirmações sobre a identidade, atributos e direitos de um sujeito para outra entidade.

O SAML é um protocolo desenhado para ser flexível e extensível, e ser usado, personalizado se necessário, por outra normalização. A *Liberty Alliance*, o projecto *Internet2 Shibboleth* e o comité OASIS *Web Services Security* (*WS-Security*), todos adoptaram o protocolo SAML como base tecnológica para diversas finalidades.

A primeira versão deste protocolo foi o SAML v1.0 lançado em Novembro de 2002. Em Setembro de 2003 foi lançado o SAML v1.1 e obteve bastante sucesso, ganhando impulso na área dos serviços financeiros, educação superior, governo e outras áreas da indústria. O SAML tem sido amplamente implementado por todos os grandes fabricantes de aplicações Web para gestão de acessos. O SAML também tem sido usado em aplicações para servidores e também muito usado em *Web Services* e em segurança.

O SAML v2.0 foi construído sobre este sucesso. O SAML v2.0 unifica a estrutura para federação de identidades definida no SAML v1.1 com as estruturas para a federação de identidades definidas no projecto *Internet2 Shibboleth* e na *Liberty Alliance*.

O protocolo SAML tráz benefícios como a:

- ***Independência da Plataforma*** – o SAML permite uma abstracção da estrutura de segurança das diferentes arquitecturas e plataformas. Um importante eixo para uma arquitectura orientada a serviço é tornar a segurança mais independente da lógica da aplicação.
- ***Experiência online para o utilizador melhorada*** – o SAML permite *Single Sign-on*, permitindo a um utilizador a autenticação num provedor de identidades e depois ter o acesso a outros provedores de serviços sem ser necessária nova autenticação. Em adição, a federação de identidades com o SAML permite uma melhor experiência personalizada, em cada um dos provedores de serviços, promovendo a privacidade.
- ***Redução dos custos administrativos para os provedores de serviços*** – o uso do SAML permite que um simples acto de autenticação (como fazer o *loggin in* com um *username* e *password*) possa ser reutilizado em múltiplos serviços reduzindo os custos de manutenção da informação contida nas contas dos seus utilizadores. Esta tarefa é transferida para o provedor de identidades.
- ***Transferência de Risco*** – o SAML passa a responsabilidade da gestão de identidades para o provedor de identidades, que será mais de acordo com o seu modelo de negócio do que do modelo de negócio de um provedor de serviço.

Como convém a uma qualquer estrutura para a comunicação de informação de identidade e segurança, o SAML tem sido aplicado de diversas maneiras, sendo as mais frequentes o *Single Sign-on* e a autorização baseada em atributos e na segurança de *Web Services*.

1.2.1 Web Single Sign-On

O *Single Sign-on* (SSO), permite a autenticação num *Web Site* e depois, sem nenhuma autenticação adicional, permitir o acesso personalizado a recurso noutro site. O SAML permite *web* SSO através da comunicação de uma afirmação autenticada do primeiro *Web Site* para o segundo, que se confiar na origem da afirmação, pode efectuar o *login* do utilizador como se este tivesse sido autenticado directamente.

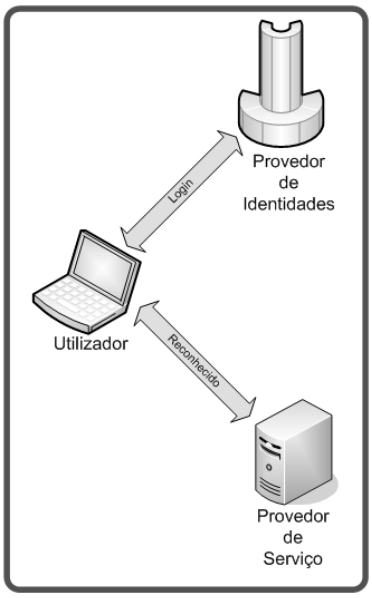


Figura 58 – Web Single Sign-On

O modelo básico de SSO é representado na Figura 58. Um utilizador autentica-se junto de um provedor de identidades e é posteriormente reconhecido (e concedido o correspondente acesso ou serviço) no provedor de serviço.

1.2.2 Autorização Baseada em Atributos

Similar ao cenário de *Web Single Sign-on*, o modelo de autorização baseada em atributos consiste na comunicação de informação de identidade do utilizador de um *Web Site* para outro *Web Site* para suportar uma determinada transacção. Contudo, esta informação de identidade pode conter características do utilizador em vez de, ou para além de, informações sobre como e quando o utilizador se autenticou. Este modelo de autorização baseado em atributos é importante quando a identidade em particular de um determinado utilizador não é importante, e não deve ser partilhada por razões de privacidade, ou insuficiente por si só.

1.2.3 Segurança em Web Services

Assertions[2] SAML podem ser usadas dentro de mensagens SOAP com o intuito de transmitir informações de segurança e de identidade num *Web Service* entre os diversos agentes. O SAML *Token Profile* especifica como é que as afirmações SAML devem ser usadas para este propósito, em conjunto com a estrutura *WS-Security*[3]. A estrutura *Identity Web Service* da *Liberty Alliance* (ID-WSF) baseou as suas especificações para usar afirmações SAML para permitir o acesso seguro e respeitando a privacidade a um *Web Service*.

[2] *Assertion* – ver secção 1.2.4
[3] WS-Security – ver secção 1.5

WS-Trust[4], uma componente da estrutura *WS-**, propôs protocolos para a troca e validação de *tokens* usados como descritos na WS-Security. Afirmações SAML são um dos formatos suportados.

1.2.4 Componentes do SAML

O SAML está definido em termos de *Assertions*, Protocolos, *bindings* e perfis.

Assertion

Uma *Assertion* é um pacote de informação que fornece uma ou mais declarações feita por uma autoridade SAML. O SAML define três tipos diferentes de declarações de *Assertions* que podem ser criadas por uma autoridade SAML:

- ***Autenticação –*** O sujeito especificado foi autenticado de um modo determinado a determinada altura. Este tipo de *Assertion* é tipicamente gerado por uma autoridade SAML que é o Provedor de Identidades, que está encarregue de autenticar os utilizadores e manter um registo de outro tipo de atributos.
- ***Atributo –*** O sujeito especificado está associado com os atributos fornecidos.
- ***Decisão de Autorização –*** O pedido para permitir o acesso ao recurso especificado por parte do sujeito especificado, foi concedido ou negado.

Figura 59 – Assertion SAML

A estrutura de uma *Assertion* é genérica, fornecendo informações que são comuns a todas as declarações contidas dentro dela. Dentro de uma *Assertion*, uma série de elementos interiores descrevem a autenticação, os atributos, a decisão de autorização ou declarações definas pelo utilizador contendo outro tipo de informações mais específicas. A Figura 59 ilustra uma afirmação SAML típica de autenticação.

[4] WS-Trust – ver secção 1.6

Protocolos

O SAML define um número de protocolos de pedidos e resposta que permitem provedores de serviço:

- Pedir a uma autoridade SAML uma ou mais afirmações (incluindo o pedido directo de uma determinada afirmação, assim como o pedido de *assertions* que correspondem a determinado critério).
- Pedir a um provedor de identidades para autenticar um *principal* e devolver a afirmação correspondente.
- Pedir que determinado identificador seja registado.
- Pedir que o uso de determinado identificador seja terminado.
- Receber uma mensagem de protocolo que foi pedida por meio de um artefacto.
- Pedir um quase simultâneo *logout* de um determinado conjunto de sessões – *Logout* único (*Single Logout*).
- Pedir que determinado identificador seja mapeado.

Bindings

O mapeamento da troca de mensagens SAML em mensagens normalizadas ou protocolos de comunicação é chamado de SAML *Protocol Binding*. Por exemplo, o SAML SOAP *Binding* define como é que as mensagens do protocolo SAML podem ser comunicadas dentro de mensagens SOAP, enquanto o HTTP *Redirect Binding* define como passar as mensagens SAML através da redirecção HTTP.

Perfis

Um perfil do protocolo SAML geralmente define constrangimentos e/ou extensões para suportar o uso do protocolo SAML em determinada aplicação, com o objectivo de melhorar a interoperabilidade removendo inevitavelmente alguma da sua flexibilidade. Por exemplo, o perfil Web Browser SSO especifica como é que as afirmações SAML de autenticação são comunicadas entre um provedor de identidades e um provedor de serviço para permitir um login único ao utilizador de um browser.

O perfil Web SSO define como se deve usar o protocolo SAML de autenticação em conjunto com diferentes combinações do HTTP Redirect, HTTP POST, HTTP Artifact e SOAP bindings.

Uma descrição mais detalhada pode ser encontrada em [7].

1.3 XACML

XACML ou *Extensible Access Control Markup Language* [8] é uma linguagem baseada em XML.

Desenhada especificamente para a criação de políticas, e para a automatização do seu uso no controlo de acessos a dispositivos e aplicações na rede.

XACML é uma iniciativa para o desenvolvimento de uma norma para sistemas de controlo de acessos e autorização. Actualmente, a maior parte dos sistemas implementa o controlo de acesso e autorização de uma forma proprietária.

Um cenário típico de controlo de acesso e autorização envolve três identidades -- um sujeito, um recurso e uma acção – e seus atributos. Um sujeito faz um pedido para realizar uma acção num determinado recurso. Por exemplo, no pedido de acesso "Permitir ao gestor de conta de um banco criar ficheiros na pasta Empréstimos no servidor de contas do banco", o sujeito é o "gestor de conta", o recurso pretendido é a "pasta Empréstimos no servidor de contas do banco" e a acção é "criar ficheiros".

Num sistema de controlo de acessos proprietário, a informação acerca destas identidades e dos seus atributos é mantida em repositórios. Estes repositórios são chamados de Listas de Controlo de Acessos (ACLs – *Acess Control Lists*). Havendo diferentes sistemas proprietários, cada um com a sua implementação de ACLs, torna-se difícil a troca e partilha de informação entre eles.

O XACML tem por objectivos criar uma forma portável e normalizada de descrever entidades de controlo de acesso e seus atributos e permitir um mecanismo de acesso de controlo mais sofisticado do que simplesmente conceder ou negar acessos.

1.3.1 Arquitectura do XACML

O XACML é composto por variados componentes descritos no diagrama da Figura 60.

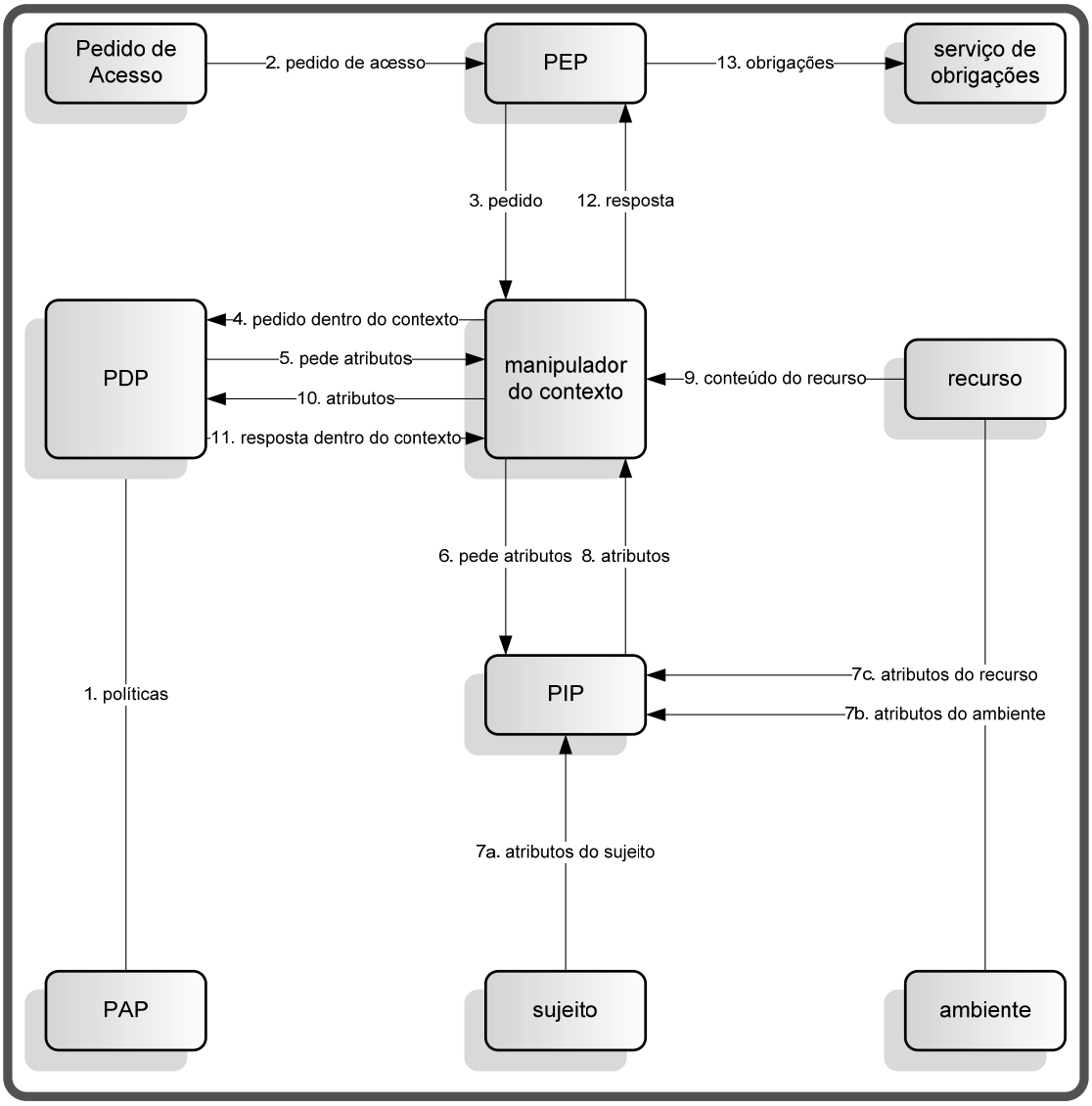


Figura 60 – Componentes da Arquitectura do XACML

Esta arquitectura opera seguindo os seguintes passos:

1. O PAP cria as políticas e conjuntos de políticas e torna-as disponíveis ao PDP. Estas políticas ou conjuntos de políticas representam por completo todas políticas para um determinado alvo.
2. Um sujeito faz um pedido de acesso ao PEP.
3. O PEP manda o pedido de acesso para o manipulador de contexto (*context handler*), no seu formato nativo, incluindo opcionalmente atributos do sujeito, do recurso, acção ou do ambiente (*environment*).
4. O manipulador de contexto constrói um XACML *request context* e envia ao PDP.

5. O PDP pede ao manipulador de contexto, caso seja disso, atributos do sujeito, recurso, acção ou ambiente.
6. O manipulador de contexto pede os atributos de um PIP.
7. O PIP obtém os atributos pedidos.
8. O PIP retorna os atributos pedidos ao manipulador de contexto.
9. Opcionalmente, o manipulador de contexto inclui o recurso no contexto.
10. O manipulador de contexto envia os atributos pedidos e (opcionalmente) o recurso pedido ao PDP. O PDP avalia as políticas de acesso.
11. O PDP envia a *response context* (incluindo a decisão de autorização ou não) para o manipulador de contexto.
12. O manipulador de contexto traduz a *response context* para o formato nativo do PEP e de seguida envia a resposta para o PEP.
13. O PEP preenche as obrigações.

Se o acesso é permitido, o PEP permite o acesso ao recurso, se não, nega-o.

1.3.2 XACML context

O XACML tem como intuito adequar-se a variadíssimos cenários de aplicação. A linguagem do *core* é isolada do cenário de aplicação pelo contexto XACML como é mostrado na Figura 61, na qual o alcance da especificação do XACML é indicada pela área sombreada. O contexto XACML é definido num esquema XML, descrevendo uma representação canónica para as entradas e saídas do PDP. Os atributos referenciados por uma instância de uma política XACML podem estar no formato de uma XPath (XML *Path Language*) sobre o contexto, ou designadores de atributos que identificam o atributo pelo sujeito, recurso, acção ou cenário e seu identificador, tipo de dados e (opcionalmente) o seu emissor.

Implementações têm que converter entre a representação dos atributos no cenário da aplicação e (SAML, J2SE, CORBA,…) e a representação dos atributos dentro do contexto XACML.

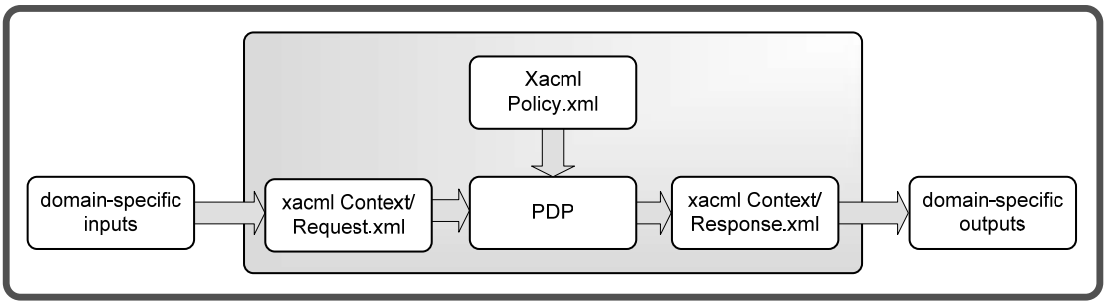


Figura 61 – Contexto XACML

1.3.3 Modelo da Linguagem de Políticas do XACML

O XACML segue um modelo de políticas que está representado na Figura 62. Os componentes principais do modelo são:

- *Rule*,
- *Policy*, e
- *Policy set*.

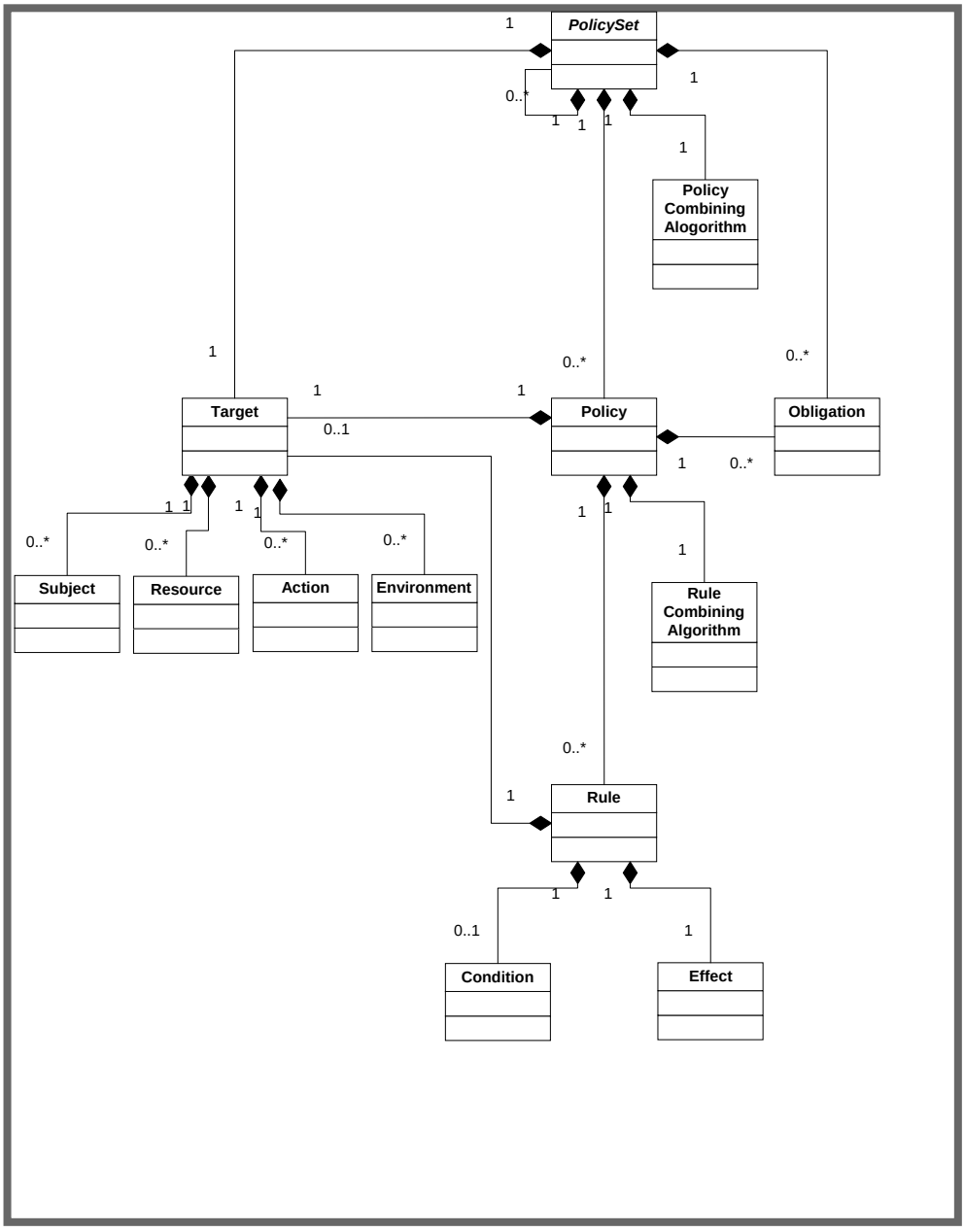


Figura 62 – modelo da linguagem de políticas

1.3.3.1 Rule

Uma *rule* é a unidade elementar de uma política. Pode existir isoladamente apenas dentro de um dos actores principais de um domínio XACML. Para haver troca de *rules* entre actores principais, as *rules* têm que estar encapsuladas numa *policy*. Uma regra pode ser avaliada com base no seu conteúdo. Os componentes principais de uma *rule* são: *target*, *effect* e *condition*.

Rule target

O componente *target* define o conjunto de recursos, sujeitos, acções e cenários aos quais a *rule* se aplica. O elemento *condition* pode estender a aplicabilidade definida pelo *target*. Se a *rule* é para ser aplicada a todas as entidades de um tipo de dados em particular, então a entidade correspondente é omitida do objectivo. Um PDP XACML verifica que as correspondências definidas no *target* são satisfeitas pelo sujeito, recurso, acção e atributos do cenário, no pedido de contexto. As definições contidas no elemento *target* são discretas para permitir que a aplicação das *rules* seja identificada eficientemente pelo PDP.

O elemento *target* pode não estar definido na *rule* e nesta a situação o *target* da rule é o que está definido na *policy* em que esta *rule* está definida.

Rule Effect

O componente *Effect* de uma *rule* indica qual a consequência de uma avaliação positiva (*true*) para a *rule*. Dois valores são possíveis: "*Permit*" e "*Deny*".

Rule Condition

O componente *Condition* representa uma expressão booleana que refina a aplicabilidade da *rule* definida no componente *target*. Pode não existir.

1.3.3.2 Policy

A partir do modelo representado pelo diagrama da Figura 62 pode ver-se que as *rules* não são trocadas entre entidades de sistema. Um PAP combina várias *rules* numa *policy*. Uma *policy* em composta por quatro componentes principais: *target*, *rule-combining algorithm-identifier*, um conjunto de *rules* e *obligations*.

Policy target

Um PolicySet, uma policy ou uma rule, todos eles contêm o elemento target que define o conjunto de subjects, resources, actions e enviroments a que são aplicadas. O target de um policyset ou de uma policy pode ser declarado pelo criador do policyset ou da policy, ou pode ser calculado a partir dos elementos target que o contêm (policyset, policy e rule).

No caso em que o elemento *target* é declarado numa *policy*, qualquer componente *rule* que tenha o mesmo *target* pode omitir o elemento *target*, visto que se este for omisso o componente *rule* herda o elemento *target* da *policy* que a contém.

Rule-combining algorithm

O elemento *Rule-combining algorithm* especifica o procedimento pelo qual os resultados da avaliação dos componentes *rules* são combinados aquando da avaliação da *policy*, isto é, o valor *Decision* colocado na resposta de contexto pelo PDP é o valor da *policy*, definido pelo *rule-combining algorithm*. Uma *policy* pode conter parâmetros que afectam a operação deste elemento.

Obligations

As *obligations* podem ser adicionadas pelo criador da policy.

Quando um PDP avalia uma *policy* contendo *obligations*, ele retorna estas *obligations* para o PEP na resposta de contexto.

1.3.3.3 Policy Set

Um Policy Set contém quatro componentes principais: target, policy-combining algorithm-identifier, um conjunto de policies e obligations.

Policy-combining algorithm

O elemento *Policy-combining algorithm* especifica o procedimento pelo qual os resultados da avaliação dos componentes *policies* são combinados aquando da avaliação do *policy set*, isto é, o valor *Decision* colocado na resposta de contexto pelo PDP é o resultado da avaliação do *policy set*, definido pelo *policy-combining algorithm*. Um *policy set* pode conter parâmetros que afectam a operação deste elemento.

Obligations

O criador de um *policy set* pode adicionar *obligations* ao *policy set*, em adição aos que poderão estar definidos nas *policies*.

Quando um PDP avalia um *policy set* contendo *obligations*, ele retorna estas *obligations* para o PEP na resposta de contexto.

Uma descrição mais detalhada pode ser encontrada em [9].

1.4 SPML

Outra norma importante para a Gestão de Identidades é o protocolo SPML (*Service Provisioning Markup Language*) [10], que tem como finalidade lidar com o provisionamento de recursos para a informação de identidades. Por exemplo, com o SPML é possível automatizar todo o mecanismo de provisionamento necessário quando uma entidade contrata um novo empregado.

A importância de uma norma para provisionamento pode não ser relevante para uma pequena entidade, mas se estivermos a pensar numa grande organização que têm centenas ou milhares de trabalhadores e têm um grande número de sistemas informáticos, aplicações e até mesmos sistemas externos à organização mas aos quais os empregados têm que ter acesso, uma norma para o provisionamento pode revelar-se de extrema importância. Numa entidade ou organização desta dimensão imensos problemas poderão surgir quando por exemplo uma nova aplicação é disponibilizada aos empregados ou apenas a parte destes, ou aquando da contratação de um novo empregado. Problemas de provisionamento podem surgir além dos já referidos a nível do acesso a sistemas, ficheiros, directorias ou até a edifícios, poderão surgir também outros respeitantes a um domínio físico, como por exemplo obter um novo portátil, telemóvel ou até um espaço de trabalho para um novo empregado. São várias as coisas que têm que ser feitas aquando da contratação de um novo empregado. Agora, se pensarmos nas coisas que também têm que ser feitas prontamente quando um empregado deixa a organização, ou quando a organização termina uma parceria com outra entidade, para garantir que o espólio da organização não possa ser usado por quem já não deve, de uma forma ilegal, verificamos que a importância do provisionamento é realmente muito elevada.

A importância de uma norma para o provisionamento ainda é mais elevada, porque hoje em dia vivemos num mundo global, em que as entidades e organizações estão ligadas e usam os diversos sistemas disponibilizados entre eles para maximizar a eficiência e produtividade dos recursos existentes. Assim uma norma para o provisionamento ajuda na gestão do provisionamento através das entidades e organizações e também de sistemas diferentes.

Resumindo, o SPML pretende:

- ***Tarefas de provisionamento automatizadas*** – Ao normalizar a linguagem torna-se mais fácil de encapsular nos vários requisitos de segurança e auditoria de um sistema de provisionamento, o SPML torna o provisionamento o mais automático possível.
- ***Interoperabilidade entre os diferentes sistemas de provisionamento*** – Diferentes sistemas de provisionamento podem interagir entre eles usando para isso uma interface SPML normalizada.

1.4.1 Componentes de um sistema de provisionamento

Um sistema de provisionamento é composto, essencialmente, por três componentes: a *Request Authority* (RA), o *Provisioning Service Point* (PSP) e o *Provisioning Service Target*, como está representado na Figura 63:

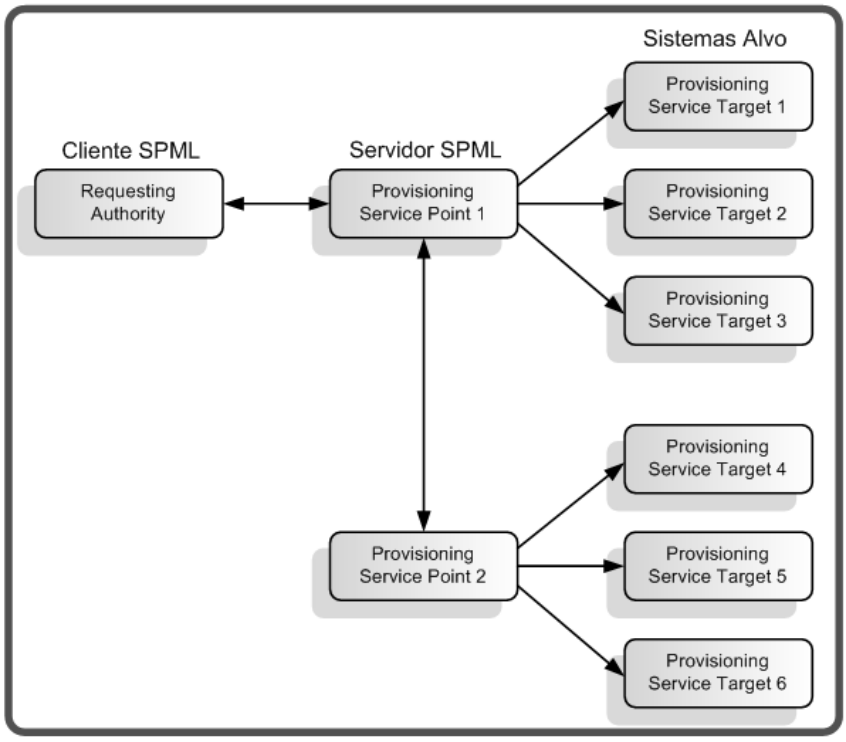


Figura 63 – Componentes de um Sistema de provisionamento

- ***Requesting Authority (RA)*** – É o cliente na arquitectura do SPML. É ele que cria documentos SPML e os envia como pedidos para o SPML *Service Point*. Estes pedidos descrevem uma operação para ser efectuada em determinados *Service Points*. Para um RA poder enviar um pedido a SPML *Service Point*, tem que existir uma relação de confiança entre o RA e o SPML *Service Point*. Um SPML *Service Point* pode actuar como um RA quando envia um pedido SPML para outro *Service Point*.
- ***Provisioning Service Point (PSP)*** – Este é o componente que recebe os pedidos do RA, processando-os e devolvendo uma resposta ao RA. Qualquer componente que receba e processe documentos SPML é chamado de *Provisioning Service Point*.
- ***Provisioning Service Target (PST)*** – Este é o software onde a acção é executada. Por exemplo, pode ser o software que armazena todas as contas de utilizador de uma organização, ou um sistema que recebe os pedidos para por exemplo a aquisição de um novo portátil para um empregado.

1.4.2 Funcionalidades do SPML

No SPML existe:

- Um conjunto de operações do core
- Um protocolo de pedido/resposta
- E uma definição do esquema de serviço.

1.4.2.1 Operações do core do SPML

As operações definidas no *core* do SPMLv2.0 são algumas operações básicas (*add*, *modify*, *delete*, *lookup*) que são aplicadas a um objecto num *target*. Existe também uma operação de descoberta (*listTargets*) num provedor.

- *listTargets* – esta operação permite ao RA determinar o conjunto de *targets* disponíveis para provisionamento e determinar que capacidades têm cada um destes *targets*.
- *add* – esta operação permite ao *RAcriar* um novo objecto num *target*, e opcionalmente ligar este objecto a um outro objecto pai.
- *modify* – esta operação permite ao RA modificar um objecto num *target*. Permite modificar um esquema definido para um objecto e/ou também os dados que um dado objecto suporta.
- *delete* – esta operação permite ao RA remover um objecto de um *target*.
- *lookup* – esta operação permite a um RA obter o XML que representa um objecto num *target*. Esta operação também obtém quais são o tipo de dados associados ao objecto.

Exemplo

Um administrador do sistema de recursos humanos de uma empresa pretende adicionar um novo empregado. Este empregado deve ter acesso ao sistema de contabilidade da empresa, que é gerido pelo departamento de contabilidade. Uma maneira deste empregado ser registado no sistema de contabilidade da empresa é o administrador do sistema de recurso humanos enviar um e-mail para o administrador do sistema de contabilidade da empresa e aguardar a resposta deste. Usando o SPML, que permite efectuar um pedido deste tipo de uma forma automática e estruturada, seria simplesmente usar a operação *add* para adicionar o novo empregado ao sistema.

O SPML permite definir operações para serviços para além das listadas acima. Permite operações chamadas de *extended operations*, que permite que clientes façam pedidos e recebam respostas com sintaxes e semânticas predefinidas.

1.4.2.2 Protocolo de pedido/resposta do SPML

Este protocolo define como os vários componentes do SPML comunicam entre eles. Este protocolo dita que um pedido feito por um cliente descreve uma ou mais operações para serem efectuadas num *Service Point* específico. Este pedidos podem ser feitos individualmente ou em grupo, e serem submetidos sincronamente ou assincronamente, ou seja, um pedido tem dois aspectos diferentes a serem definidos:

- Modelo de execução do pedido – síncrono ou assíncrono
- Combinação ou agrupamento de pedidos – individualmente ou em grupo

Modelo de execução do pedido

Um pedido processado sincronamente é bloqueante para o RA. O RA após enviar o pedido, fica à espera para receber a resposta. No caso dos pedidos processados assincronamente, o processo torna-se mais complexo. O RA envia cada pedido com um identificador único (ID) e não espera pela resposta. O ID é guardado pelo PSP enquanto processa o pedido e devolve o ID em conjunto com a resposta ao RA. Nesta situação, pode haver o interesse do cliente (RA) em saber qual é o estado do pedido. O ID também é útil nesta situação. O identificado único (ID) do pedido também é usado para controlar e gerir pedidos pendentes no PSP.

Para controlar e gerir os pedidos, o SPML fornece duas operações:

- *StatusRequest* – esta operação permite aos clientes saberem qual o estado de um pedido executado assincronamente.
- *CancelRequest* – esta operação permita ao RA cancelar a execução de pedido assíncrono pendente.

Combinação ou agrupamento de pedidos

Um RA pode enviar um pedido individualmente ou em grupo. Em qualquer dos casos, os pedidos podem ser executados de uma forma síncrona ou assíncrona. Os pedidos individuais são processados da forma como foi visto antes. O processamento dos pedidos feitos em grupo tem três aspectos importantes que se têm que definir:

- ***Resultado do processamento –*** como é que cada resposta de cada pedido do grupo é enviada ao cliente.
- ***Tipo de processamento –*** como é que este pedidos são processados, se são processados em paralelo ou sequencialmente.
- ***Tratamento de erros –*** como é que os erros são tratados se um ou alguns pedidos do grupo não for processado com sucesso.

Resultado do processamento: Múltiplas operações SPML são agrupadas e enviadas como um único pedido *BatchRequest*. Os resultados do processamento pelo PSP são enviados como uma única resposta *BatchResponse* baseada na sua posição original no *BatchRequest*, isto é, a primeira resposta do *BatchResponse* corresponde ao primeiro pedido do *BatchRequest*, a segunda resposta ao segundo pedido e assim sucessivamente.

Tipo de processamento: O SPML tanto suporta o processamento de uma forma sequencial como em paralelo. O RA pode determinar como é que este processamento é feito. No caso de um processamento sequencial, os pedidos são processados pela ordem em que estão no grupo de pedidos. No processamento paralelo, o servidor pode efectuar os processamentos em qualquer ordem. Mas em qualquer dos casos a resposta é baseada na sua posição original no pedido.

Tratamento de erros: O SPML fornece duas opções para tratar os casos em que o processamento de um pedido falha. Essas duas opções são: retomar e sair. O RA especifica qual destas duas opções prefere na altura em que o pedido é enviado. Quando a opção escolhida é a de retomar, o insucesso do processamento de um pedido não afecta a execução dos restantes pedidos, a posição original é mantida. Quando a opção é sair, o insucesso do processamento de um pedido faz com que o servidor termine a execução dos restantes pedidos e todos os pedidos não executados são marcados como falhados.

1.4.2.3 Definição do esquema de serviço do SPML

O esquema de serviço permite aos clientes interrogar um servidor SPML pelos detalhes das operações que este suporta.

O esquema de serviço do SPML é baseado no esquema W3C XML e adiciona-lhe uma definição da classe objecto e um modelo de partilha de atributos.

No exemplo seguinte pode-se ver que os atributos são definidos num esquema XML e também que os atributos estão agrupados segundo a definição classe objecto[5].

```
<schema majorVersion="1" minorVersion="0">
    <providerIdentifier providerIDType="urn:oasis:names:tc:SPML:1:0#URN">
        <providerID>urn:oasis:names:tc:SecF</providerID>
    </providerIdentifier>
    <schemaIdentifier
schemaIDType="urn:oasis:names:tc:SPML:1:0#GenericString">
        <schemaID>PersonSchema</schemaID>
    </schemaIdentifier>
    <attributeDefinition name="FullName"
                    description="Full name of the employee joining."/>
    <attributeDefinition name="email" description="E-mail address."/>
    <attributeDefinition name="description" description="Description."/>
    <attributeDefinition name="project" description="Project assigned
to."/>
    <objectclassDefinition name="employee" description="Sample
employee.">
        <memberAttributes>
            <attributeDefinitionReference name="FullName"
required="true"/>
            <attributeDefinitionReference name="email" required="true"/>
            <attributeDefinitionReference name="description"/>
            <attributeDefinitionReference name="project"
required="true"/>
        </memberAttributes>
    </objectclassDefinition>
</schema>
```

Os clientes usam o protocolo pedido/resposta descrito acima para obter a informação do esquema. A operação fornecida pelo SPML para obter a informação do esquema chama-se *ShemaRequest*. *SchemaRequest* serve para obter um esquema de provisionamento específico. Estes esquemas são obtidos usando identificadores, especificamente, o identificador do esquema e o identificador do provedor. Se nenhum identificador de esquema for fornecido, todos os esquemas suportados pelo provedor são devolvidos. O identificador do provedor é o único identificador do provedor, e pode ser uma organização ou um indivíduo que é responsável por um conjunto de operações. O identificador de um esquema é o único identificador de um esquema que um provedor oferece. Esta garantia de unicidade é dada no contexto do provedor.

[5] Classe objecto é uma forma de agrupar os atributos e de os referenciar através de um nome.

O SPML é um protocolo que se encaixa perfeitamente no contexto da Gestão de Identidades. Torna o provisionamento de identidades automático e inter-operável entre diversas entidades e organizações, factor muito importante em Gestão de Identidades.

Uma descrição mais detalhada pode ser encontrada em [11]

1.5 WS-Security

O *WS-Security* [12] foi a primeira especificação a ser desenvolvida aproveitando as capacidades de extensão do SOAP. O *WS-Security* tem como propósito definir formas de proteger trocas de mensagens SOAP e fornecer um meio de transporte para informação relacionada com segurança.

O XML quando foi criado não fornecia qualquer tipo de mecanismo de segurança que garantisse qualquer tipo de integridade ou confidencialidade. O XML teve um enorme sucesso e houve então necessidade de criar mecanismos de segurança. Foram criados duas normas, o XML *Signature* [59] e o XML *Encryption* [60], que descrevem formas de aplicar criptografia a documentos XML. Estas normas são bastante versáteis, permitindo que apenas algumas partes do documento fossem encriptadas e usassem chaves diferentes.

O *WS-Security* descreve como aplicar o XML *Signature* e o XML *Encryption* a um tipo especial de documentos XML, as mensagens SOAP. Uma mensagem SOAP, como visto na anteriormente, tem uma estrutura que se adapta perfeitamente a este modelo. A mensagem pode ser modificada de acordo com a operação pretendida, por exemplo, substituindo o *body* por dados encriptados, enquanto o *header* SOAP tem a descrição da encriptação que foi feita. O receptor da mensagem analisa o conteúdo do cabeçalho (um *WS-Security Soap header* neste caso), e descobre que o corpo da mensagem foi encriptado usando um determinado algoritmo e uma determinada chave. Se o receptor tiver a chave correspondente, pode reverter o processo e desencriptar a mensagem. A assinatura da mensagem funciona de uma forma análoga.

Exemplo:

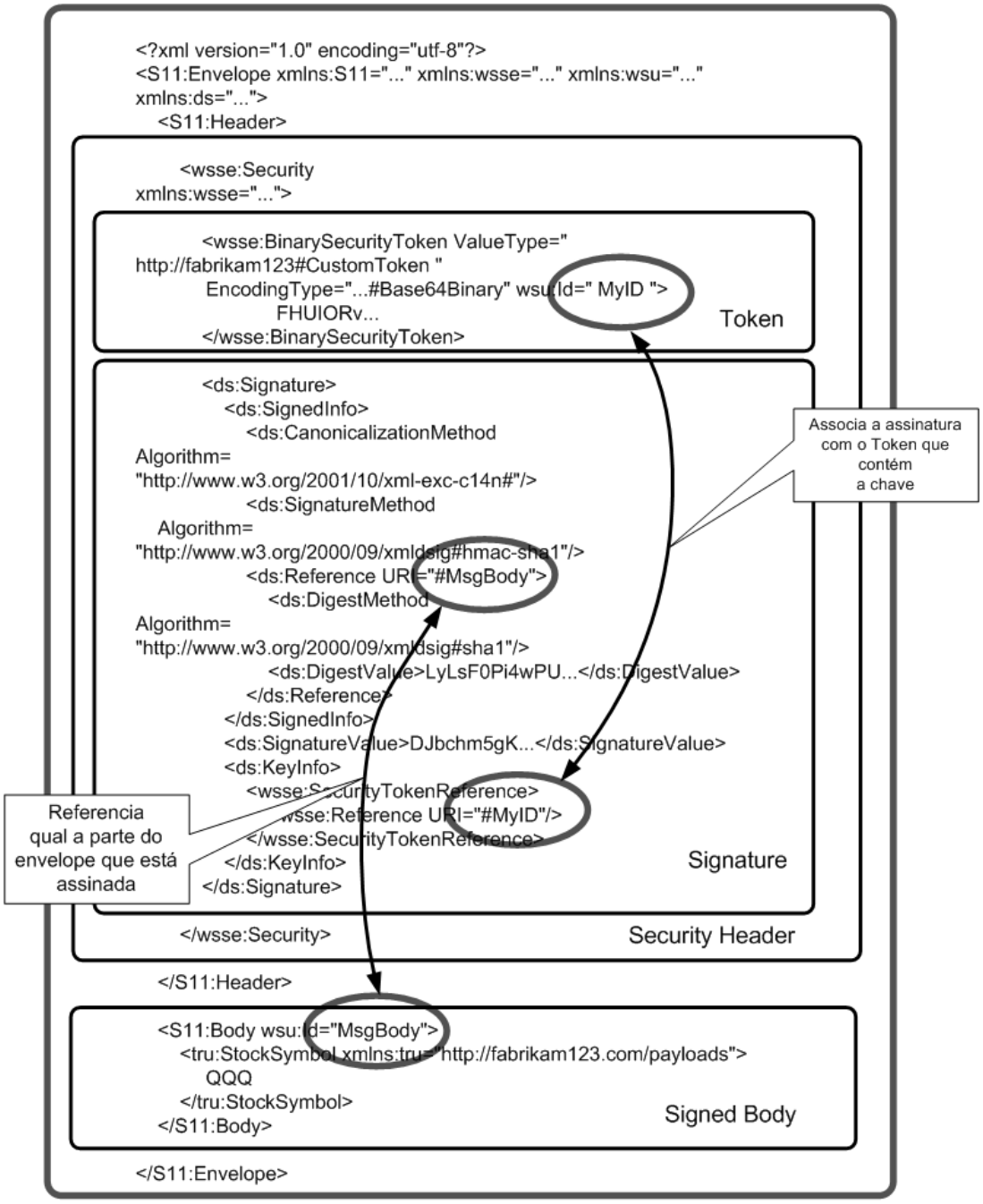


Figura 64 – Exemplo de uma mensagem SOAP com o corpo assinado usando WS-Security

O *WS-Security* não introduz nenhum tipo novo de criptografia. O *WS-Security* tem que acomodar as tecnologias de segurança existentes e promove a interoperabilidade entre elas. O *WS-Security* tem que ser capaz de encriptar e assinar mensagens SOAP usando as tecnologias disponíveis aos utilizadores: X.509, SAML, Kerberos, *username* e *passwords*, etc, assim como tecnologias que possam aparecer no futuro. Sendo assim, não pode existir nenhum tipo de dependência explícita na especificação do *WS-Security*. A especificação assume que o material criptográfico é enviado num *WS-Security security token*. É neste *token* que são enviadas as várias definições de encriptação e assinatura da mensagem. Existem especificações separadas para cada tipo de tecnologia, chamadas de *token profiles*. Cada *token profile* descreve como derivar um *WS-Security token* de uma tecnologia de segurança existente. Por exemplo um *Web Service* pode exigir que as mensagens que recebe têm que vir assinadas, mas não especifica o tipo de chave que quer que seja usada. Um emissor pode assinar usando um *security token* derivado de um certificado X.509 enquanto outro pode usar outro usando um *security token* derivado de um Kerberos *ticket*. Desde que o receptor para verificar a assinatura tenha o certificado correspondente, ou fazendo parte do domínio Kerberos, consegue fazer a verificação com sucesso. É esta separação que permite que uma nova tecnologia que seja desenvolvida no futuro possa ser integrada sem qualquer tipo de problemas e que permite que actualmente diferentes utilizadores possam utilizar a tecnologia para a comunicação – SOAP.

Uma descrição mais detalhada pode ser encontrada em [13].

1.6 WS-Trust

Como visto, o *WS-Security* suporta qualquer tipo *security token* desde que os requisitos definidos no *token profile* sejam aplicados. Qualquer tecnologia de Autenticação baseada na emissão de *tokens* descreve a sua maneira de como um cliente pode obter um *token*. Por exemplo operação de Autenticação usando Kerberos ou SAML é feita de uma forma completamente diferente. O *WS-Trust* [14] generaliza a operação de emissão de *tokens* a *WS-Security tokens*, isto é, o *WS-Trust* estende o *WS-Security* com métodos para emissão, renovação, e validação de *security tokens* de uma forma que é igual para qualquer plataforma e tecnologia. O *WS-Trust* permite modelar, de uma forma tecnologicamente agnóstica, as operações necessárias para obter *tokens*.

O *WS-Trust* introduz um tipo especial de *Web Service* que é o *Security Token Service* (STS). Basicamente, a função do STS é "transformar" *WS-Security Tokens*. Um *token* entra e outro *token* sai.

Exemplo:

Um determinado utilizador quer invocar um determinado *Web Service*. O *Web Service* especifica nas suas políticas por razões de segurança que apenas aceita pedidos se forem seguros por um tipo *WS-Security tokens*, por exemplo, *WS-Security tokens* baseados em SAML contendo um determinada afirmação acerca do utilizador. O utilizador pode pedir a um STS para lhe emitir o SAML *token* que ele precisa para invocar *Web Service*. O pedido ao STS é feito enviando-lhe uma mensagem especial, cujo formato é especificado no *WS-Trust*, chamada de *Request for Security Token* (RST). O RST contém, para além de outras coisas, uma descrição do tipo de *token* que o utilizador está a pedir ao STS para ele emitir. O STS, contudo, não emite *tokens* a qualquer um. Como o SAML *token* tem que ter uma afirmação acerca do utilizador, o STS tem que ter a certeza de que é o utilizador que está a fazer o pedido, ou seja, tem que ter a certeza que o RST é proveniente daquele utilizador específico. Assim, o utilizador tem que segurar a mensagem RST de uma forma que convença o STS de que foi ele que a enviou. Em termos de *WS-Security*, isto quer dizer que o utilizador tem que segurar o RST usando algum tipo de *security token*. Por exemplo, o utilizador pode usar um *WS-Security security token* derivado de Kerberos.

Quando o STS recebe o RST, examina-o e verifica se ele vinha devidamente seguro. Se tudo estiver de acordo com o esperado, o STS considera a identidade do utilizador como verificada procede à geração do SAML *token* pedido. Este novo *token* é enviado para o utilizador noutra mensagem especial, cujo formato está especificado no *WS-Trust*, chamada de *Request* for *Security Token Response* (RSTR). Quando o utilizador recebe a RSTR, extrai o SAML *token* e usa-o para invocar o *Web Service*. O *Web Service* recebe o SAML *token* e verifica que ele contém uma afirmação acerca do utilizador, que satisfaz as suas políticas de segurança.

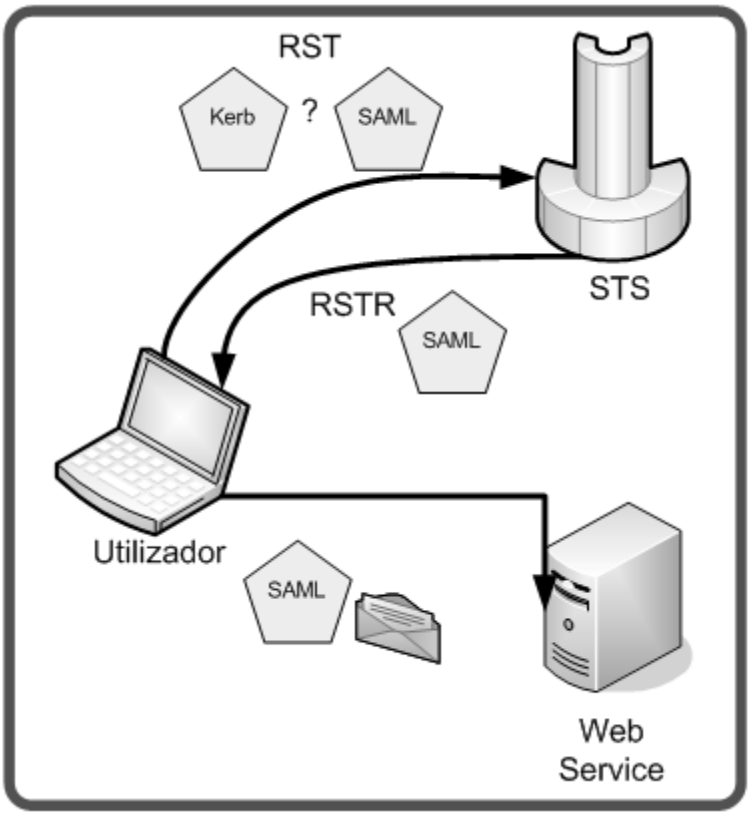


Figura 65 – representação do WS-Trust em acção

Para que isto tudo funcione tem que haver uma relação de confiança entre o *Web Service* e o STS. Mas de qualquer modo o *Web Service* tem que ter uma prova de o SAML *token* foi realmente emitido pelo STS que ele confia. O STS assina com a sua chave privada todos os *tokens* que envia, assim qualquer um que conheça a chave pública do STS pode verificar a sua proveniência. Para além disso o utilizador demonstra que o SAML *token* foi enviado realmente para ele pela sua capacidade de usar um *token* que vinha dentro da mensagem RSTR que recebeu do STS para segurar o seu pedido ao *Web Service*.

O *WS-Trust* define entidades e mensagens para a emissão de *WS-Secutity tokens* via *Web Services*. A especificação define entre outras situações mas é esta que demonstra a sua aplicação na Gestão de Identidades.

Uma descrição mais detalhada pode ser encontrada em [15].

1.7 WS-MetadataExchange

Os *Web Services* usam meta data para descrever aos seus clientes o que eles precisam de saber para poderem interagir com ele. Por exemplo, o *WS-Policy* [61] descreve as capacidades, requisitos e as características gerais dos *Web Services* e o WSDL [62] descreve as operações sobre as mensagens, protocolos de rede e os endereços usados como *endpoint* pelo *Web Service*. O

WSDL e o *WS-Policy* fornecem uma forma de descrever o *Web Service* ao mundo. À medida que as transacções baseadas em Web Services foram ficando mais complexas, houve necessidade de definir como obter a meta data do *Web Service* de uma forma normalizada e programática. O *Ws-MetadataExchange* [16] é um protocolo que cumpre esse propósito. Permite ao cliente inquirir o *Web Service* e obter a sua informação em meta data, tipicamente as suas políticas.

Uma descrição mais detalhada pode ser encontrada em [16].

2 Lista de Claims suportada pelos *Self-issued Cards* ou *Personal Cards*

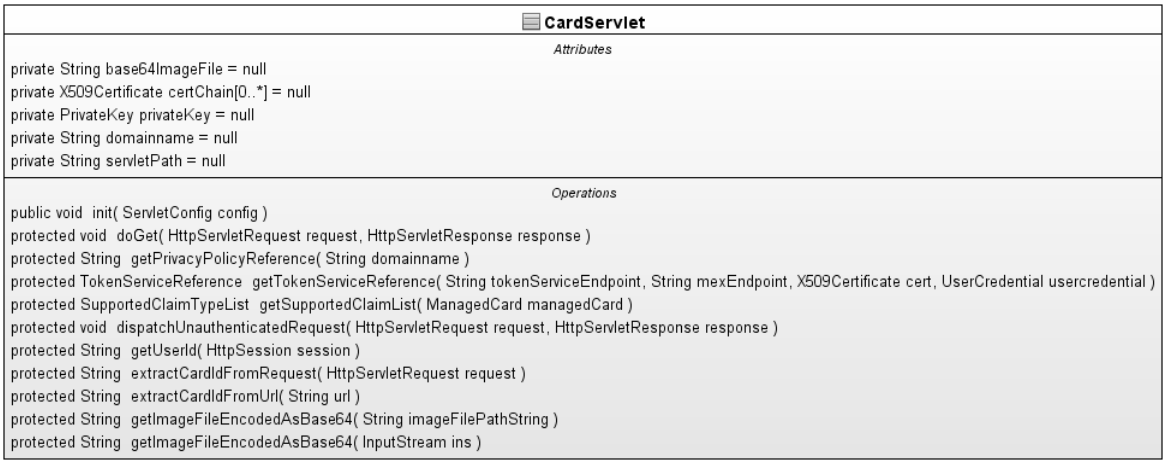
O URI http://schemas.xmlsoap.org/ws/2005/05/identity/claims assim como os seguintes estão definidos no ISIP 1.5 [72].

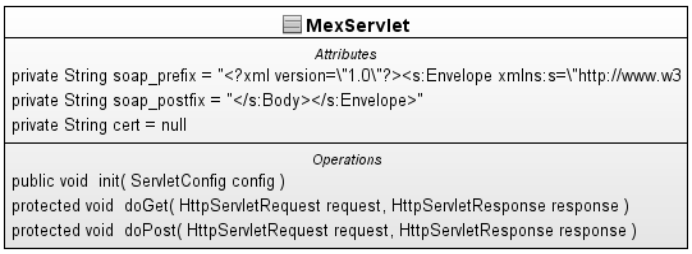
Claim name	Data Type	Description	URI
givenname	xs:string	(givenName no RFC 2256) Nome preferido ou primeiro nome do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
surname	xs:string	(sn no RFC 2256) Sobrenome ou nome de família do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
emailaddress	xs:string	(mail no inetOrgPerson) Endereço de Email prefeirdo para o campo "To:" de um email a ser envidado para o sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
streetaddress	xs:string	(street no RFC 2256) Endereço da Rua, é uma componente da informação do endereço do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/streetaddress
locality	xs:string	(l no RFC 2256) Localidade, é mais uma componente da informação do endereço do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/locality
stateorprovince	xs:string	(st no RFC 2256) Estado ou Província, é também outra componente da informação do endereço do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/stateorprovince
postalcode	xs:string	(postalCode no X.500) Código Postal, é também outra componente da informação do endereço do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode
country	xs:string	(country no RFC 2256) País do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/country
homephone	xs:string	(homePhone no inetOrgPerson) Telefone primário ou de casa do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/homephone
otherphone	xs:string	(telephoneNumber no X.500 Person) Telefone secundário ou do trabalho do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/otherphone
mobilephone	xs:string	(mobile in inetOrgPerson) Número de telemóvel do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/mobilephone
dateofbirth	xs:date	Data de nascimento do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth
gender	xs:token	Sexo do sujeito: 0 – não especificado, 1 – masculino, 2 – feminino.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/gender
Privatepersonal	xs:base64binary	Um private personal identifier (PPID) identifica o sujeito num determinado	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier

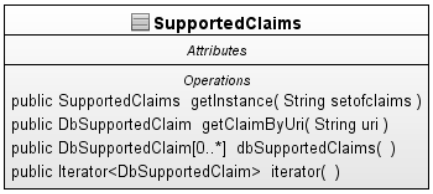
identifier		provedor de serviço de forma a manter a sua privacidade.	
webpage	xs:string	O URL da página pessoal do sujeito.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/webpage

3 My Identity Provider v1.5 – Classes Principais

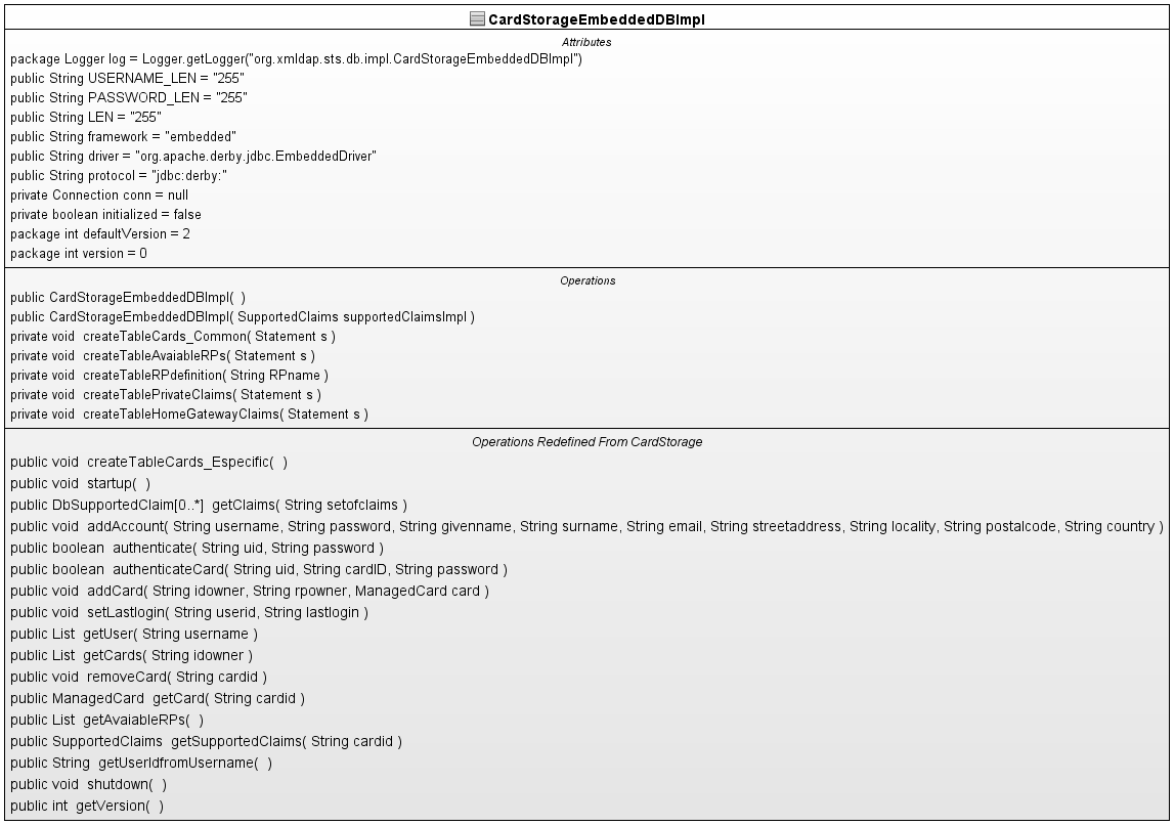
De seguida apresenta-se o conteúdo (métodos e atributos) das Classes mais importantes do MyIdP v1.5

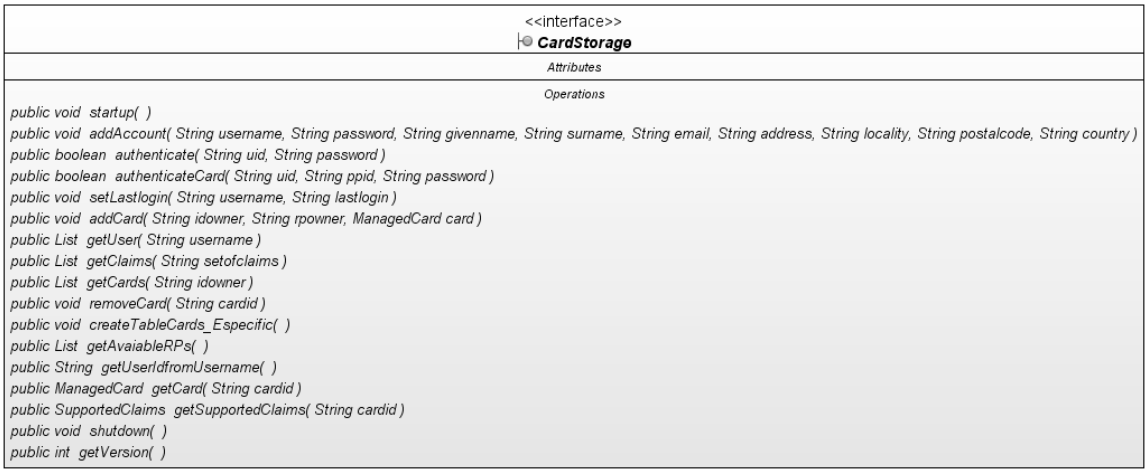


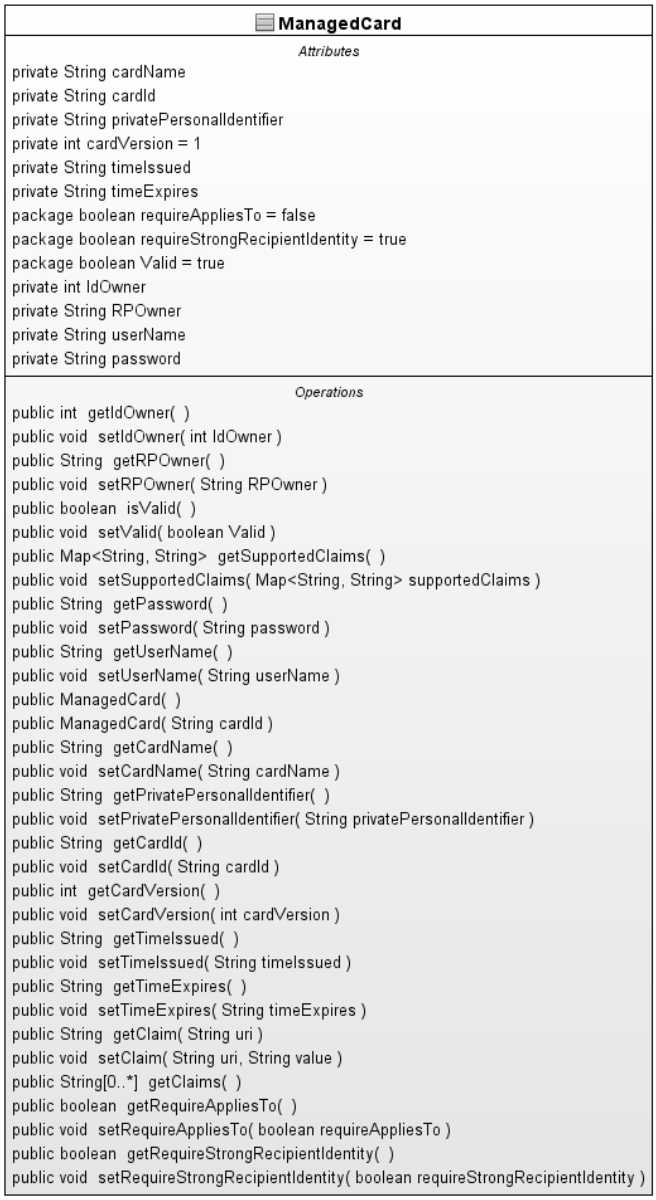





STSServlet
Attributes
package Logger log = Logger.getLogger("org.xmldap.sts.servlet.STSServlet")
package RSAPrivateKey key = null
package X509Certificate cert = null
package String domain = null
private String servletPath = null
Operations
public void init(ServletConfig config)
protected void doPost(HttpServletRequest request, HttpServletResponse response)
ControllerHelper
Attributes
package String address
private PrivateKey privateKey = null
Operations
protected String jspLocation(String page)
public ControllerHelper(HttpServletRequest request, HttpServletResponse response)
public String submitregistrationMethod( )
public String loginMethod( )
public void setlastLogin( )
public String registerMethod( )
public String removecardMethod( )
public String createcardMethod( )
public String createcardfromRPMethod( )
Operations Redefined From HelperBase
public void copyFromSession(Object sessionHelper)
protected void doGet( )
protected void doPost( )


CardStorageEmbeddedDBImpl
Attributes
package Logger log = Logger.getLogger("org.xmldap.sts.db.impl.CardStorageEmbeddedDBImpl")
public String USERNAME_LEN = "255"
public String PASSWORD_LEN = "255"
public String LEN = "255"
public String framework = "embedded"
public String driver = "org.apache.derby.jdbc.EmbeddedDriver"
public String protocol = "jdbc:derby:"
private Connection conn = null
private boolean initialized = false
package int defaultVersion = 2
package int version = 0
Operations
public CardStorageEmbeddedDBImpl( )
public CardStorageEmbeddedDBImpl(SupportedClaims supportedClaimsImpl)
private void createTableCards_Common(Statement s)
private void createTableAvaiableRPs(Statement s)
private void createTableRPdefinition(String RPname)
private void createTablePrivateClaims(Statement s)
private void createTableHomeGatewayClaims(Statement s)
Operations Redefined From CardStorage
public void createTableCards_Especific( )
public void startup( )
public DbSupportedClaim[0..*] getClaims(String setofclaims)
public void addAccount(String username, String password, String givenname, String surname, String email, String streetaddress, String locality, String postalcode, String country)
public boolean authenticate(String uid, String password)
public boolean authenticateCard(String uid, String cardID, String password)
public void addCard(String idowner, String rpowner, ManagedCard card)
public void setLastlogin(String userid, String lastlogin)
public List getUser(String username)
public List getCards(String idowner)
public void removeCard(String cardid)
public ManagedCard getCard(String cardid)
public List getAvaiableRPs( )
public SupportedClaims getSupportedClaims(String cardid)
public String getUserIdfromUsername( )
public void shutdown( )
public int getVersion( )


<<interface>>
CardStorage
Attributes
Operations
public void startup( )
public void addAccount(String username, String password, String givenname, String surname, String email, String address, String locality, String postalcode, String country)
public boolean authenticate(String uid, String password)
public boolean authenticateCard(String uid, String ppid, String password)
public void setLastlogin(String username, String lastlogin)
public void addCard(String idowner, String rpowner, ManagedCard card)
public List getUser(String username)
public List getClaims(String setofclaims)
public List getCards(String idowner)
public void removeCard(String cardid)
public void createTableCards_Especific( )
public List getAvaiableRPs( )
public String getUserIdfromUsername( )
public ManagedCard getCard(String cardid)
public SupportedClaims getSupportedClaims(String cardid)
public void shutdown( )
public int getVersion( )


ManagedCard
Attributes
private String cardName
private String cardId
private String privatePersonalIdentifier
private int cardVersion = 1
private String timeIssued
private String timeExpires
package boolean requireAppliesTo = false
package boolean requireStrongRecipientIdentity = true
package boolean Valid = true
private int IdOwner
private String RPOwner
private String userName
private String password
Operations
public int getIdOwner( )
public void setIdOwner(int IdOwner)
public String getRPOwner( )
public void setRPOwner(String RPOwner)
public boolean isValid( )
public void setValid(boolean Valid)
public Map<String, String> getSupportedClaims( )
public void setSupportedClaims(Map<String, String> supportedClaims)
public String getPassword( )
public void setPassword(String password)
public String getUserName( )
public void setUserName(String userName)
public ManagedCard( )
public ManagedCard(String cardId)
public String getCardName( )
public void setCardName(String cardName)
public String getPrivatePersonalIdentifier( )
public void setPrivatePersonalIdentifier(String privatePersonalIdentifier)
public String getCardId( )
public void setCardId(String cardId)
public int getCardVersion( )
public void setCardVersion(int cardVersion)
public String getTimeIssued( )
public void setTimeIssued(String timeIssued)
public String getTimeExpires( )
public void setTimeExpires(String timeExpires)
public String getClaim(String uri)
public void setClaim(String uri, String value)
public String[0..*] getClaims( )
public boolean getRequireAppliesTo( )
public void setRequireAppliesTo(boolean requireAppliesTo)
public boolean getRequireStrongRecipientIdentity( )
public void setRequireStrongRecipientIdentity(boolean requireStrongRecipientIdentity)